

CLIENT/SERVER PROGRAMMING WITH CORBA

THIS
ISSUE: **MORE!**
OpenDOC

September/October 1994
Display Until 10/31/94

os/2 Developer

THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT

**WRITING OS/2 PM
CONTROLS IN C++**

**BUYER'S GUIDE:
OBJECT-ORIENTED
DEVELOPMENT
TOOLS**

**A Distributed
Object
Road Map**

**A 32-Bit
List Box
Replacement**

**Wrapping
in Smalltalk**

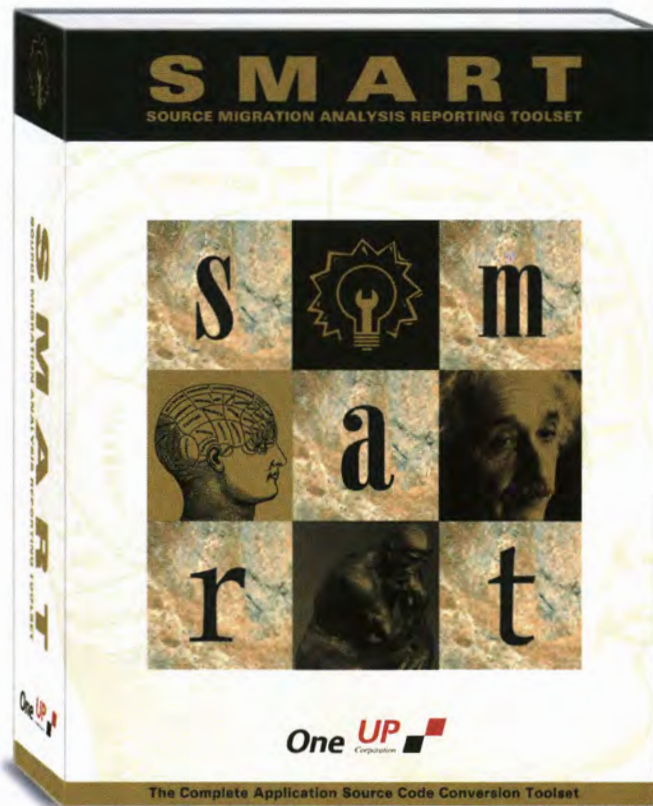
**PIECE
TOGETHER
POWERFUL
OBJECTS
WITH OS/2**

U.S. \$9.95 Vol. 6 No. 5



A Miller Freeman Publication

Converts Source Code To Native 32-Bit OS/2.



NO MIRRORS! NO KIDDING!

For more information, call 800-678-01UP or FAX 214-620-9626.

One UP
Corporation

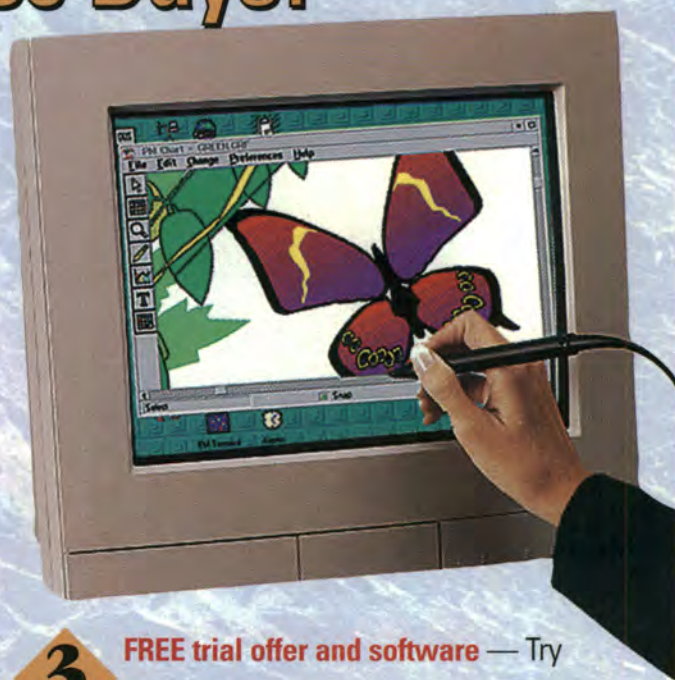
1603 LBJ Freeway, Suite. 200 • Dallas, TX 75234

Circle Reader Service Number 1

Three Reasons to Try the Future of Business Computing™ Free for 30 Days!

PenD/rect™ for OS/2™

a product of FTG Data Systems



1 PenDirect is the most direct link between you and your application

— PenDirect's one-to-one cursor interaction means long-term computer use that's simple and effortless. The pen's precise and quick input maximizes cursor control to eliminate "mouse dyslexia." You'll find your natural hand/eye coordination beats fumbling with an awkward mouse any day. Light pens also eliminate the tedious wrist movements associated with mice that cause fatigue and repetitive strain injuries.

2 Add pen support instantly to all your applications

— PenDirect works great with all of your OS/2 software, with no extra programming or customization. PenDirect fully supports all OS/2 2.x, Pen for OS/2, Windows and DOS apps — at any screen resolution. Everything you need is included in the package, no special monitor is required, and PenDirect lets you continue to use your mouse.

3 FREE trial offer and software

— Try the pen free for 30 days — after 30 days you may return the pen or keep it and pay just \$298. No strings, no catches. You must be convinced PenDirect is the best input device for OS/2 or return it and owe nothing.

And for a limited time, get our Pen for OS/2 driver absolutely free with your paid order — that's a \$49 value FREE!

**For a free 30-day evaluation unit,
or more information, call:**

800-962-3902

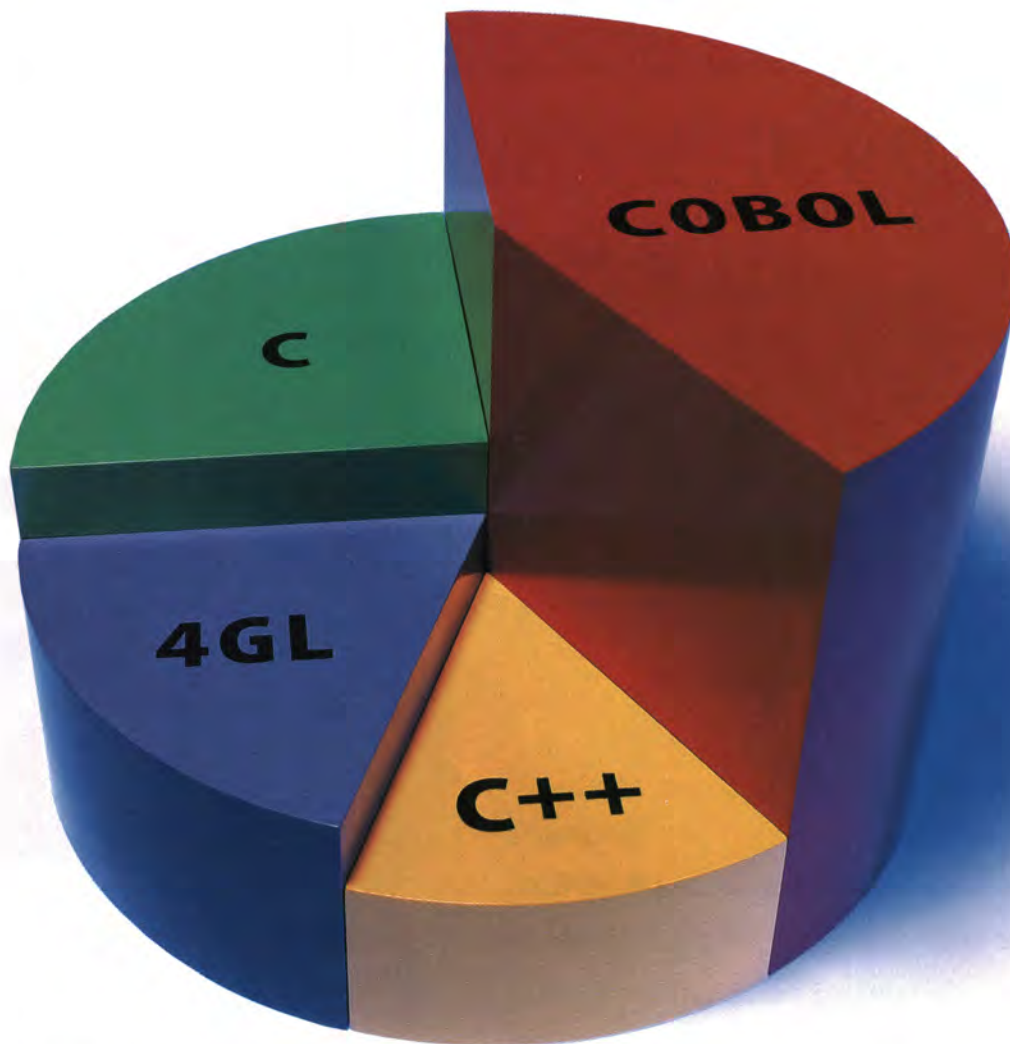
714-995-3900 • FAX 714-995-3989

**FTG DATA
SYSTEMS**

8381 Katella Avenue
Stanton, California 90680

© 1994 FTG Data Systems. All rights reserved. Screen shot courtesy of Micrografx, Inc. Image created using PM Chart v1.1.

PenDirect is a trademark of FTG Data Systems. OS/2 is a trademark of International Business Machines. All other trademarks are property of their respective owners.



With all the hype surrounding business programming, one thing has been conspicuously absent: the facts.

No matter how you slice it, COBOL means business programming.

According to the Gartner Group 80% of all programmers today use COBOL.

That's right, COBOL. The language that some, shall we say, "unenlightened" industry pundits think has gone the way of punch-cards and vacuum tubes. The reality is, COBOL is more vital to business today than it ever was and at Micro Focus, we know why.

Versatility—We've made COBOL the most "open" system in the industry. Micro Focus COBOL programmers can target applications to DOS, Windows, OS/2 and literally hundreds of UNIX® environments (something that

current C, C++ and 4GLs only aspire to do). We've also added communications facilities and data access capabilities that make COBOL the ideal choice for developing client/server or cross-platform applications.

Performance—We've made COBOL scream. Applications written in Micro Focus COBOL run as fast as anything written in C or C++ and faster than most 4GL-based applications, too.

Productivity—We've made COBOL more productive than C, C++ and 4GLs by adding advanced tools like Animator® for streamlining source code debugging as well as syntax extensions to handle graphical user interfaces,

file and record locking and other important functions. We've also brought the latest development facilities like Dialog System Professional™, our rapid application development tool for Windows and OS/2® Revolve™, our comprehensive system-wide analysis tool, and even object orientation, with our Object COBOL Option™.

Those are just a few reasons why programmers insist on Micro Focus COBOL for business application development. To learn more, call 1-800 MF COBOL and ask for our white paper: "COBOL vs. C."

Micro Focus: The past, present and future of programming.

MICRO FOCUS®

Micro Focus and Animator are registered trademarks of Micro Focus Ltd. Dialog System Professional, Revolve and Object COBOL Option are trademarks of Micro Focus. Other trademarks are property of their respective companies. Graph based on Sentry Market Research Study AD/CASE 1993/94.

Circle Reader Service Number 3

OS/2 Developer

SEPTEMBER/OCTOBER 1994

THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT



EDITOR'S COMMENTS

Gary Kildall, CP/M and OS/2 5



GUI CORNER

Object of Desire 6



CORPORATE STUDY

Life After Outsourcing 18



PROGRAMMING INSIDER

Objective OS/2 Decisions 24



OBJECT-ORIENTED PROGRAMMING

A Distributed Object Road Map for Developers 31

By Cliff Reeves, Robert Orfali, and Dan Harkey

OpenDoc Basic Anatomy 101 40

By Robert Tycast

Client/Server Programming with CORBA 46

By Robert Orfali and Dan Harkey

Hints for VisualAge Wrappers 54

By Robert Lindsay

Writing OS/2 PM Controls in C++ 61

By Eric Snell and Lori Ruffing



TOOLS

Object-Oriented Development Tools Buyer's Guide 68



PRODUCT WATCH

New Tools for OS/2 77



Programmer's Paradise®

Call for a
FREE
Catalog!



Watcom C/C++ V10.0 by Watcom

Comprehensive C and C++ development system for 32-bit DOS, Windows NT, Win 32s, OS/2 2.x, and Novell NLMS, and 16-bit DOS and Windows 3.x. Delivers productivity and performance, combining state-of-the-art compiler technology with a new integrated development environment (IDE) and comprehensive set of tools. Includes advanced GUI debugger, C++ class browser, profiler, and more. Support for C++ templates, exception handling and Microsoft Foundation Class libraries (MFC).



List: \$199 Ours: \$189* FAXcetera #: 1683-0012

* While supplies last.

key 01SO94

Watcom VX•REXX V2.1 by Watcom

NEW!
Version 2.1

Watcom VX•REXX is an easy to use visual development environment for creating applications that leverage the capabilities of OS/2 2.x and exploit its CUA '91 graphical user interface. VX•REXX combines a project management facility, visual designer and an interactive source-level debugger to deliver a very approachable and highly productive visual development environment. With the visual designer, you can create an attractive graphical interface using VX•REXX's full suite of CUA '91 interface objects, quickly customize their properties, and easily attach REXX procedures to the objects. Drag-and-drop programming lets you write event routines without typing any code.

Special Introductory Price Only \$99
Client/Server Edition Only \$299
FAXcetera #: 1683-0016

key 03SO94



GpfREXX by Gpf Systems, Inc.

WYSIWYG visual OS/2 PM programming with REXX. Point and click to create PM design using the full CUA '91 control set (e.g., NoteBooks, Containers, etc.), as well as Multi-Media Controls. GpfREXX supports standard OS/2 features like Drag-and-Drop and Multi-Tasking as well as the extended features, including: Multi-Media, SQL (DB/2 including Remote services), and advanced communications (APPC, CPI-C, EHLLAPI). No royalties. (Compatible with Gpf 2.1)

List: \$129 Ours: \$119
GpfREXX & GpfTools: List: \$274 Ours: \$253
FAXcetera #: 3582-0005

key 05SO94

Halo Imaging Library for OS/2 by Lifeboat Publishing

HALO Imaging Library for OS/2 enables you to incorporate powerful image processing into your applications with over 100 C/C++ imaging functions. This advanced toolkit provides a full range of functionality including the ability to display, enhance, print, manipulate, and save most popular digital image formats. Special effects such as sharpen, blur, warp, color conversion, scaling, and more are also included.



List: \$795 Special Price: \$456
FAXcetera #: 1045-0015

key 07SO94

Circle Reader Service Number 4

CA-REALIZER by Computer Associates

The first truly accessible approach to creating GUI applications. Using a structured superset of BASIC, you can quickly and easily create complete OS/2 and Windows applications without adding third party DLLs or learning about events, state-based systems, application cooperation, device control, and all of the other new techniques and complexities introduced by the other GUI programming systems. If you can create a program in DOS, you can create a program in OS/2 and Windows with CA-REALIZER!



List: \$247 Ours: \$79* FAXcetera #: 1004-0008

* While supplies last.

key 02SO94

Smalltalk/V for OS/2 by Digital

Tap into the power of 32-bit technology and the feature set of IBM's OS/2 with Smalltalk/V for OS/2, the first 100% pure object-oriented language for the Client/Server operating platforms of tomorrow. Smalltalk/V for OS/2 is a pure object-oriented development language that delivers up to twice the speed with 32-bit power, and gives you the portability to run on OS/2, Win32 or Macintosh.



List: \$995 Ours: \$795 FAXcetera #: 1469-0001

key 04SO94

PM Designer by Vac Nat Software

PM Designer is a visual, extensible GUI toolset that makes PM development easy. Create fully functional multi-threaded applications with toolbars, status bars, multiple windows, and more in minutes with the interactive design environment. Visually design and test your application's behavior, and quickly generate C/C++ source code, resource, IPF, and makefiles. Extensible design supports any class library, compiler, or language.



List: \$99 Ours: \$92 FAXcetera #: 1015-2901

key 06SO94

To order call: 800-445-7899
Corporate Developer Division: 800 441-1511
FAX: 908 389-9227
International: 908 389-9228
Customer Service: 908 389-9229
Programmer's Paradise Deutschland:
Tel.: 08121/79073
FAX: 08121/76566
Programmer's Paradise Italia:
Tel.: 39-2-967-00409
FAX: 39-2-967-02855

For more information on the
products featured on these pages call
FAXcetera®: (908) 389-8173

Programmer's Paradise®

1163 Shrewsbury Avenue
Shrewsbury, NJ 07702-4321

* Prices subject to change without notice.
* Call for shipping charges/return policies.



EDITOR

Dick Conklin

EXECUTIVE EDITOR

Nicole Freeman

MANAGING EDITOR

Marta Dils

EDITORIAL ASSISTANT

Diane Anderson

EDITORIAL ARTIST

Peter Altenberg

VICE PRESIDENT/**GROUP DIRECTOR**

Regina Starr Ridley

PUBLISHER

Cathy Passage

ADVERTISING SALESAngela Barnett
(415) 905-4983Holly Meintzer
(212) 626-2275Kristin Morgan
(212) 626-2498**MARKETING MANAGER**

Susan McDonald

ART DIRECTOR/MARKETING

Christopher H. Clarke

MARKETING ASSISTANT

Larra Dutton

ADVERTISING PRODUCTION**COORDINATOR**

Tom Marzella

CIRCULATION DIRECTOR

John Rockwell

CIRCULATION MANAGER

Lucinda Formyduval

SUBSCRIPTION INQUIRIES

U.S.: (800) WANT-OS2

Foreign: (708) 647-5960

CHAIRMAN OF THE BOARD

Graham J.S. Wilson

PRESIDENT/CEO

Marshall W. Freeman

EXECUTIVE VICE PRESIDENT

Thomas L. Kemp

VICE PRESIDENT, PRODUCTION

Andy Mickus

VICE PRESIDENT, CIRCULATION

Jerry Okabe



BY DICK CONKLIN

Editor's Comments

Gary Kildall, CP/M and OS/2

This summer we lost a pioneer in PC operating systems, Digital Research founder Gary Kildall. I was privileged to work briefly with Gary back in 1980, when I joined IBM's PC development team. At the time, I was running Gary's CP/M operating system on my Apple II computer. CP/M was the king of the hill back then, the industry standard in desktop operating systems. DOS existed only as SCP 86-DOS, a CP/M-like clone developed for the new 8086/8088 16-bit processors by Seattle Computer Products.

Most people assumed that CP/M would be the operating system of choice for the new IBM PC. Almost nobody had heard of DOS. Of course, existing CP/M applications would still have to be updated and recompiled for the PC's new processor, no matter which operating system was used.

Gary liked to call himself a hacker and preferred to leave the marketing and legal concerns to his staff. Scheduling and contractual problems caused a snag in the negotiations, and tiny Microsoft, led by a young and aggressive Bill Gates, seized the opportunity. SCP 86-DOS was quickly acquired, and work began on porting early applications like VisiCalc and EasyWriter to the new machine. CP/M-86 arrived on the scene several months later, but it was too late.

Today, OS/2 developers face an exciting

future. The PowerPC chip and symmetrical multiprocessing (SMP) are driving hardware technology forward, while DSOM, Taligent, and OpenDoc are doing the same for software. Only OS/2 is so well positioned to take advantage of the future, and OS/2 applications will lead the way.

Gary Kildall was a great programmer and pioneer, but he will always be remembered for that one giant missed opportunity. We will miss him.



Dick Conklin

Dick Conklin

Contact	CompuServe	Internet
Dick Conklin (Editorial)	76711,1005, or OS2DF2 Forum	os2mag @vnet.ibm.com
Miller Freeman (Circulation)	71572,341	71572,341 @compuserve.com
Cathy Passage (Publisher)	71673,1163	csp @mfi.com
Marta Dils (Managing Editor)		mdils @mfi.com

OS/2 Developer: Vol. 6, No. 5, September/October 1994

Subscription information: U.S.: One year (six issues), \$39.95. Canada, Mexico, and international surface mail, add \$16 per year for postage. Canadian GST no. 124513185. International air mail, add \$30 per year for postage. Foreign orders must be accompanied by payments in U.S. funds. For new orders and customer service, call (800) WANT-OS2 (800-926-8672) or (708) 647-5960 (fax: 708-647-0537). IBM employees and branch office customers can subscribe to OS/2 Developer through IBM Mechanicsburg's Systems Library Services (SLSS) using OS/2 Developer's order number: G362-0001. POSTMASTER: Please send address changes to OS/2 Developer, P.O. Box 1079, Skokie, IL 60076-8079. Allow eight weeks for subscriptions to begin. OS/2 Developer (ISSN 1073-0729) is published bimonthly by Miller Freeman Inc., 600 Harrison Street, San Francisco, CA 94107, (415) 905-2200. Application to mail at second-class postage rates is pending at San Francisco, CA and additional mailing offices.

OS/2 Developer has applied for BPA membership.

© Copyright 1994 by Miller Freeman Inc. PRINTED IN THE U.S.A.

in

Miller Freeman, Inc.
A MEMBER OF THE UNITED NEWSPAPERS GROUP



In this issue's column, we continue our discussion of a 32-bit replacement list box. This time we'll show you how to write a C++ wrapper for the 32-bit list box and integrate it into the IBM C Set++ User Interface Class Library. **By MARK BENGE and MATT SMITH**

Object of Desire



Mark A. Bengé



Matt Smith

As we continue our journey in writing a 32-bit replacement list box, we would like to examine an area of particular interest to many, to which this issue is dedicated: object-oriented programming. We will develop a C++ class, `Listbox32`, which is a wrapper of the 32-bit list box control. We will design `Listbox32` for integration into the IBM C Set++ User Interface Class Library (ICLUI).

Although hints are provided, the C Set++ documentation is incomplete concerning the design and implementation of a custom control using the C Set++ ICLUI. We will code `Listbox32` and document key areas in this article that will show you not only how to design and implement a C++ custom control but also how to replace the underlying Presentation Manager control and inherit its support and functionality from ICLUI, all without using the "hairball" treatment (if you're a cat owner, you know what we mean)!

The 32-bit list box is a custom control that can be used as a replacement for the Presentation Manager list box. Since it is a replacement and not a new control, wrapping it presents a few challenges we'll show you how to overcome. The following sections describe the steps required to design and code `Listbox32`.

TO SUBCLASS OR TO WRITE FROM SCRATCH?

The C Set++ list box class, `IListBox`, is a wrapper for the Presentation Manager list box control. Although we have added extensions to the 32-bit list box, many of the member functions within `IListBox` can be reused without any modifications. Writing from scratch means reimplementing everything. Guess what? Yes, you guessed it: we're lazy, so we're taking the path of least resistance. However, this approach does present some other hurdles, which we'll show you how to clear. Therefore, our task is to show you the necessary steps to create a subclass of `IListBox`, which we'll denote as `Listbox32`. We will design our subclass to accommodate a snug fit with the ICLUI hierarchy.

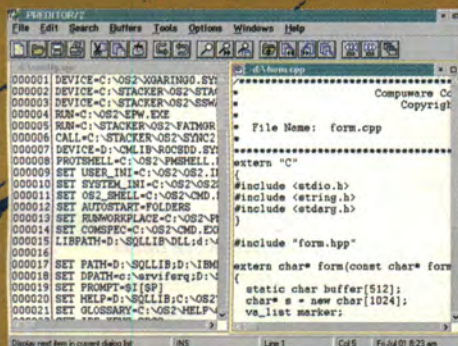
SO YOU WANTED TO BE IN CONSTRUCTION?

Before building the `Listbox32` object, we need to satisfy the underlying Presentation Manager requirements—remember, `IListBox` uses the `WC_LISTBOX` predefined window class when creating the underlying Presentation Manager window, whereas we will be using the "ListboxWindow" private window class. To satisfy the Presentation Manager requirements, we have defined a static

The eagle
has landed...

PREDITOR/2

THE EDITOR FOR UNTAMED PROGRAMMERS



on "planet OS/2". Introducing PREDITOR/2.

Your source code has never fallen prey to a more powerful editor. PREDITOR/2 is for developers who don't have the time to deal with rigid programming environments, who must change facilities to fit their exact requirements and who need to dramatically extend editing functions to attain peak productivity. This editor stalks code with the speed and versatility of a true predator.

HIGHLIGHTS:

- CUA-compliant GUI
- Multiple document interface
- BRIEF, VI, EMACS, ISPF, CUA emulation
- WorkFrame/2 integration
- Built-in compiler support
- Extensive customization facilities
- Powerful C-like extension language
- Unlimited Undo and Redo
- Powerful search and replace functions
- No limits on file sizes, buffers or windows

Special Introductory Price... \$149.00 • MSRP \$249.00

To order PREDITOR/2 or to receive a brochure, call now. 1-800-535-8707 or fax: 1-810-737-7119

PREDITOR/2 is a trademark of Compuware Corporation. All other company or product names are trademarks of their respective owners.
© 1994 Compuware Corporation.



Circle Reader Service Number 5



Tales from the Trenches (Building C++ DLLS)

In C, exported names are the same as the functions they represent. However, in C++ the names of the external references are mangled to support function overloading. What this all boils down to is more work to generate a DLL.

Two methods can be employed to generate C++ DLLs:

- Use of `_Export` or the `#pragma export` in the source code
- CPPFILT, which is the .DEF file building tool.

We chose to use the second method to generate XLISTBOX.DLL. We filed a number of required steps to generate the DLL:

1. Compile all of the modules with the `/Ge-` option.
2. Run the CPPFILT utility on all the .OBJ files using the `-b -p` options, redirecting the output to a .DEF file.
3. Prefix the .DEF file with the following statements:

```
LIBRARY XLISTBOX INITINSTANCE TERMINSTANCE
DESCRIPTION 'C++ Wrapper for 32-Bit Replacement Listbox'
PROTMODE
DATA MULTIPLE NONSHARED READWRITE LOADONCALL
CODE LOADONCALL
EXPORTS
```

4. Link the DLL.
5. Create the import library.

Those of you who are familiar with this method will notice that we skipped the tempinc steps and the editing step to exclude undesirable symbols. The tempinc steps were skipped because a tempinc subdirectory does not exist. Also, we're lazy, so we're giving you all of the symbols CPPFILT generates, including the kitchen sink.

Building a C++ DLL can quickly become a monotonous chore that C++ developers have to do over, and over, and over, and over again. Our recommendation is that this process be automated and that the automation take place in the IBM WorkFrame/2 product.

function named `createListBox`. Why static? We haven't constructed the `ListBox32` object yet, so we must use a static public function that we defined within the scope of `ListBox32` that can be called prior to construction of the object. Figure 1 shows the definition of the `ListBox32::createListBox` function. Note how we intermix C and

C++ code in this routine to create the list box, and note the provision for extended styles. Lest we forget, the registration of our window class, `"ListBoxWindow"`, occurs here, which essentially makes it a private class. If you would like to make it a public class, see the sidebar "Unraveling the Great SYS_DLLS—OS2.INI File Mystery

in An Object of Many Colors: Using Custom Controls Within A Workplace Object article," (November/December 1993).

Once we have created the 32-bit list box control, we can construct the `ListBox32` object. Since `ListBox32` is a subclass of `IListBox`, we need to derive from `IListBox` in our `ListBox32` constructors. In the derivation, we pass the window handle, `IWindowHandle`, returned by `WinCreateWindow` to the `IListBox` constructor that takes the window handle of an existing list box control. After construction, we can tap into the existing public and protected `IListBox` member functions via inheritance. Consequently, a nice side effect of this operation is that we convince the `IListBox` member functions that they are still processing a `WC_LISTBOX`-based list box instead of ours.

Before we discuss extended styles, let's examine the constructors we have provided. In the preceding paragraph, we discussed the constructor that takes many of the arguments you would normally pass to the `WinCreateWindow` API. As illustrated in the source, these arguments are exposed using ICLUI nomenclature.

A second type of constructor supports dialogue templates. It accepts as arguments a parent window pointer for the dialogue and a control identifier of the 32-bit list box control. The last type of constructor accepts the handle of an existing Presentation Manager window. These constructors have special requirements. The two primary ones are that the input window must be validated (that is, it must exist), and the window class, `"ListBoxWindow"`, must be verified for the input window. Those of you who are familiar with ICLUI design will recognize that these three constructors are consistent with those provided with the controls ICLUI supports.

PRESTO! A PRACTICAL CLIENT/SERVER SOLUTION!

Introducing GLS-Presto 2.0.

Now you can build state of the art OS/2 and Windows applications for hundreds and even thousands less.

Presto is the only client-server development tool which generates OS/2 and Windows applications in COBOL, C and C++, for one low price.

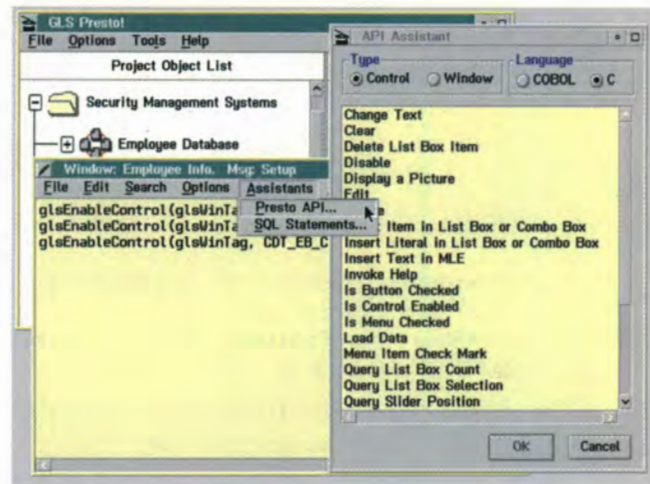
Presto is designed to help you become a client/server magician as quickly as possible without compromising on power, flexibility and security.

With Presto there is no need to invest in proprietary technology or suffer through an extensive learning curve. Presto allows you to take advantage of COBOL, C, C++ and SQL while effectively isolating you from the complexities of client/server development.



Presto's intuitive, easy to use integrated development environment will have you performing client/server tricks in no time at all. Design, prototype, generate, test and debug without ever leaving the IDE.

Presto makes the difference between OS/2 and Windows disappear, and lets you take advantage of the best of what they have to offer. For example, Windows MDI support is included for OS/2 and OS/2 control types may be used within Windows. Simply define your application, then select the target environment when you are ready to generate. Presto does the rest.



- Create complex applications in a minimum of time and run them under OS/2 and Windows.
- No knowledge of GUI or SQL programming required. The Presto code assistants are always ready to serve.
- Supports RAD methodology with instant proto-typing.
- Generated applications are quick and compact.
- Works with all major COBOL, C and C++ compilers.
- Incredibly fast. Applications generate in seconds.
- No compromises. MDI, tool bars, status bars, window geometry, custom controls and more.
- Use any DBMS or file access method compatible with your compiler.
- Fully networkable. No cumbersome object management required.
- Multi-media capability built in. Display GIF, TIF and Bitmap images dynamically.
- Absolutely no royalties or run-time fees.

Presto can appear on your computer for only



There is no better value.

To make some magic of your own, give us a call now at (219) 481-5809. Or, place an order via CompuServe, our mailbox is 73762,114.


```

/*-----|
| ListBox32::ListBox32|
| |
| Static function that creates the 32-Bit list box.|
|-----*/
HWND ListBox32::createListBox32(unsigned long    ulId,
                                HWND             pParent,
                                HWND             pOwner,
                                const RECT        &rectInit,
                                const Style       &style,
                                const ExtendedStyle &extendedStyle)
{
    /*-----|
    /* Assertions on input parms|
    /*-----|
    IASSERT(pParent!=0);

    /*-----|
    /* Register the 32-Bit list box class|
    /*-----|
    if (!RegisterListBox( IThread::current().anchorBlock() ))
        ITHROWGUIERROR( "fRegisterListBox" );

    /*-----|
    /* Build the control data structure for the 32-Bit list box|
    /*-----|
    LISTBOXCDATA ctlData;
    memset(&ctlData, 0, sizeof(LISTBOXCDATA));
    ctlData.cb = sizeof(LISTBOXCDATA);
    ctlData.ulVersion = LBV_110;
    ctlData.vdata.lbcd1_1.flExtStyles = extendedStyle.asUnsignedLong();

    /*-----|
    /* Create the 32-Bit list box|
    /*-----|
    HWND hWndLB = WinCreateWindow( pParent->handle(),
                                   "ListBoxWindow",
                                   0,
                                   style.asUnsignedLong(),
                                   rectInit.left(),
                                   rectInit.bottom(),
                                   rectInit.width(),
                                   rectInit.height(),
                                   (pOwner==0) ? IWindowHandle(0) :
                                                           pOwner->handle(),
                                   HWND_TOP,
                                   ulId,
                                   &ctlData,
                                   0 );

    if (hWndLB == 0)
    {
        ITHROWGUIERROR(IString("WinCreateWindow: Id=") +
                        IString(ulId) +
                        IString(" Class=ListBoxWindow"));
    }
    return( hWndLB );
}

```

Figure 1. ListBox32::createListBox function

Watcom™ VX•REXX™

Visualize the Power of OS/2

NEW CLIENT/SERVER EDITION

- Easily bind objects directly to your database
 - Quickly connect to your database
- Visually generate and test your SQL query
- Rapidly create more than a dozen chart types with over 150 display options

PLUS

All of the Powerful Features
of the New VX•REXX
Version 2.1

NEW

VX•REXX VERSION 2.1

- Over 2 dozen objects, including CUA'91 Containers, Notebooks, Pop-up Menus and more
- Integration and control of existing applications through DDE, keystrokes or REXX API's
- Easy to learn event-driven programming model with complete on-line documentation
- Context-sensitive help
- Powerful drag-and-drop programming techniques simplify programming
- Develop professional multi-threaded, multi-windowed and drag-and-drop enabled applications
- Code reusability through section sharing
- Include OS/2 style help and hints in your applications
- Advanced interactive source-level debugger
- System Object Model (SOM) based object manager
- Package your application as an EXE or PM macro for royalty-free distribution
- Integrated console window support simplifies migration of existing REXX programs

Watcom VX•REXX is an award winning, easy to use visual development environment for creating OS/2 applications with rich graphical user interfaces. VX•REXX combines a project management facility, visual designer and an interactive source-level debugger to deliver a very approachable and highly productive visual development environment.

Design Applications Visually

Create rich graphical applications quickly and easily using the visual design environment. With the visual designer, you can graphically create CUA'91 Presentation Manager interface objects, quickly customize their properties, and easily attach REXX procedures using powerful drag-and-drop programming techniques.

Integrated Development Environment

Build, test and debug your application without leaving the development environment. Then package your application as an EXE file or PM macro for royalty-free redistribution. The power of the integrated development environment and debugger can also be used with your existing REXX applications.

Powerful Open Environment

Enjoy the simplicity of event-driven programming together with the global editing capabilities essential for professional project management. Watcom VX•REXX is open and extensible through IBM's object-oriented System Object Model (SOM) technology. You can access all standard REXX API's including DB2/2, because VX•REXX is based on the OS/2 2.x standard system REXX.

Experts Agree... "Using VX•REXX for development is like driving a Porsche:

It's fast, it's compact, everything's in the right place, and it makes us look good, too."

Peter Coffee, PC WEEK, May 2, 1994.

"(VX•REXX) applications can be multithreaded and REXX is probably the easiest language in which to learn the OS/2 thread model."

Software Development, November, 1993.

"VX•REXX is a great tool; it's fun and productive."
PC Techniques, Dec/Jan. 1994.



for OS/2



February 8, 1994
Watcom VX•REXX, Version 1.01



"The best new development tool of 1993"

Watcom VX•REXX

Version 2.1.....**Special Price \$99**
Client/Server Edition.....**Special Price \$299**

1-800-265-4555

Watcom

A Powersoft Company

Watcom International 415 Phillip Street, Waterloo, Ontario, Canada. N2L 3X2 Phone: (519) 886-3700 Fax: (519) 747-4971

*Prices and specification are subject to change without notice. Price does not include freight and taxes where applicable. Prices quoted in US dollars. Watcom, the Lightning Device, and VX•REXX are trademarks of Watcom International Corporation. Other trademarks are the properties of their respective owners. ©Copyright 1994 Watcom International Corporation.

Circle Reader Service Number 7

STYLISH EXTENSIONS

In "A Musical List Box" (May/June 1994), we discussed extended styles and exposed the interface through the control data structure. Why? Basically, we ran out of bits in the base Presentation Manager style interface. C Set++ ICLUI provides a nested bit flag definition class that is used to provide a C++ object-oriented interface to the Presentation Manager styles. Since we are subclassing `IListBox`, we inherit the definitions and support for `IListBox::Style`. Therefore, we need to concern ourselves only with providing support for the extended styles.

We have defined a nested bit flag definition class, `ExtendedStyle`, which we will use to support our check box extended style. As we add new extended styles, we can easily add support for those to the `Listbox32` class, based upon the current extended style framework. Since we are using the control data structure interface to pass the extended styles to the 32-bit list box (shown in the module `driver.c` that was part of the `lstbx2.exe` source

code package for the May/June 1994 issue), we will continue to use that interface, as illustrated in the code in Figure 1.

We simply obtain an unsigned long representation of the extended style and store it in the `flExtStyles` field of the control data structure before passing it to `WinCreateWindow`.

IT'S IN THE MAIL

Control notifications are processed as part of the `WM_CONTROL` message. The function `dispatchHandlerEvent` in our handler class, `Listbox32Handler`, looks much like a `WM_CONTROL` case statement you would encounter in a window proc written in C. As shown in the source, you still need to verify the window class, as notification IDs overlap between controls, and you need to look for notifications of interest. The primary difference between C and an ICLUI implementation is that ICLUI encapsulates the notification via a virtual function that may be overridden in a handler subclass to provide the required behavior. Therefore, when you would like to

receive your mail, er notification, you would simply override the function, which is still less expensive than purchasing a stamp!

TOO HOT TO HANDLE

C++ handlers in ICLUI are similar to window subclass procedures in C. Window subclass procedures are typically one large switch and case block with individual case statements for messages whose default behavior is overridden. ICLUI encapsulates messages as events and isolates the processing of an event or related events in a handler, such as the keyboard handler, `IKeyboardHandler`. Virtual handler functions are then provided to expose elements of the event that may be overridden to alter the default processing. Figure 2 is a simple illustration of the message and event flow within a traditional C application and an ICLUI application, respectively. However, this figure is not concise, as we are trying to limit our discussion to the essentials.

Let's back up a little and look at the notification events that

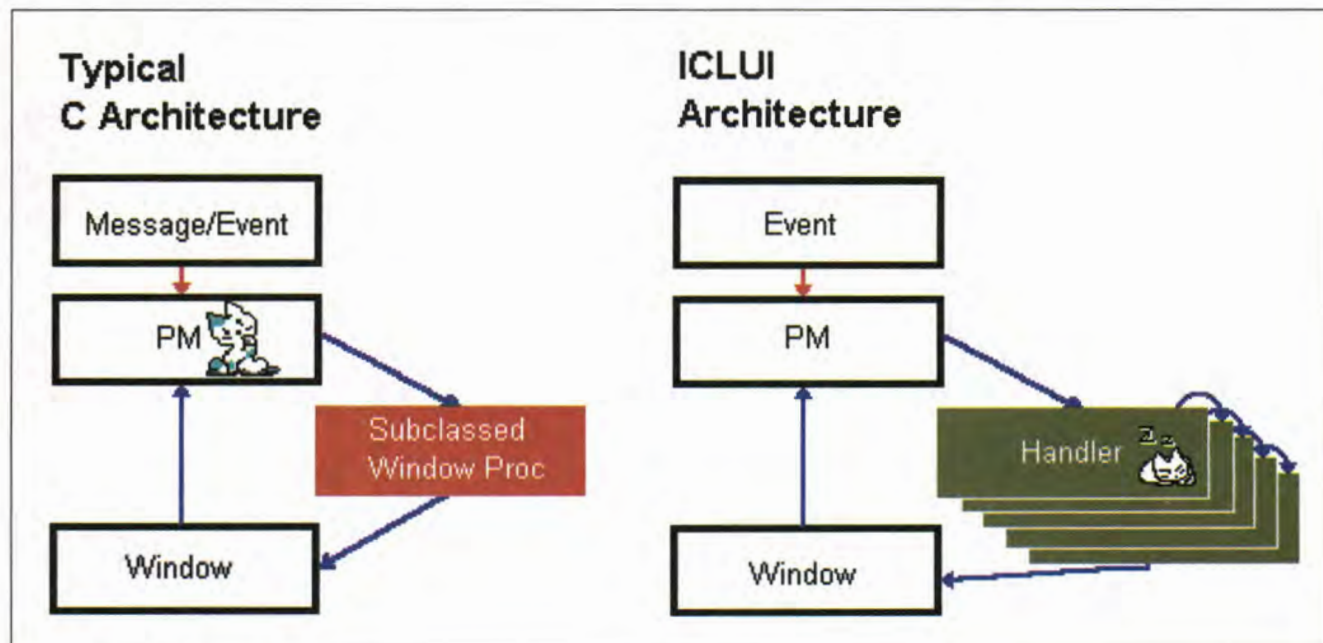
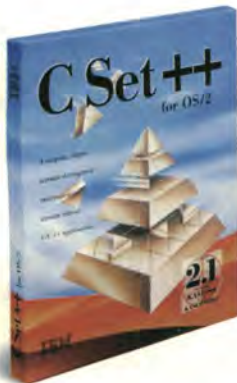


Figure 2. C vs. ICLUI event flow

Now, all the C/C++ development tools you need come in one handy package.



Bonus Bundle: C Set ++ CD-ROM, OS/2 for Windows, KASE:Set and Experience C++...just \$249*!



IBM C Set ++™ Version 2.1 is the most comprehensive C/C++ development package you can buy for OS/2®. Period.

C Set ++ can dramatically slash your programming time with an intuitive graphical interface and world-class optimization. Your applications are quick to build, fast to execute and easier than ever to modify.

C Set ++ brings you the most complete set of class libraries

and tools available for OS/2 development. It includes:

- A User Interface Class Library and Collection Classes
- Visual Debugger and Execution Trace Analyzer
- Class Browser and WorkFrame/2™ Version 2.1
- The IBM Developer's Toolkit for OS/2

...everything you need to make C/C++ development faster, easier than ever before. And where C Set ++ stops, our 24 hour comprehensive IBM service and support begins.

Now, when you purchase the C Set ++ CD-ROM, you will also receive OS/2 for Windows™, the KASE:Set™ GUI-builder and the revolutionary Experience C++ tutorial... all for just \$249*.

KASEWORKS™ KASE:Set provides an introductory level GUI-builder, speeding the pace of your developments... and your learning.

Experience C++™ is the only leading-edge multimedia tutorial that describes every aspect of C++. To order your C Set ++ Bonus Bundle call our toll-free order hotline today!



In the U.S. call
1 800 342-6672

\$249*

Your dealer may sell for less.

For the IBM C Set ++ for OS/2 CD-ROM Version 2.1.

CD-ROM with hardcopy documentation is available for \$279*, 3.5" disks with hardcopy documentation are available for \$309*.

North America:

Canada 1-800-465-7999 Ext. 690
Mexico 627-1028
U.S. 1-800-342-6672

Asia:

Australia 008-805-109
Hong Kong 852 825-6268
India 91-80-526-7117

Indonesia

62-21-520-9500 Ext. 1360
Japan 0120-04-1992
Korea 82-2-781-6555
Malaysia 03-717-7788
Philippines 815-4119
Thailand 66-2-273-4444
Taiwan 886-2-776-7878

Europe:

Austria 0660-109
Belgium 02-225-33-33
Denmark 8030-4545
Finland 90-459-6900
France 05-03-03-03
Germany 0130-812177
Greece 30-1-328-1342
Italy 1670-17001
Ireland 353-16603744

Netherlands

030-384040
Norway 66-99-93-00
South Africa 011-224-2000
Spain 900-100-400
Switzerland 01-436-6233
Sweden 08-793-4004
U.K. 0329-242728

South America:

Argentina 319-6480

© 1994 International Business Machines Corporation.

® IBM and OS/2 are registered trademarks of International Business Machines Corporation.

™ C Set ++ and Experience C++ are trademarks of International Business Machines Corporation.

KASEWORKS and KASE:Set are trademarks of KASEWORKS Inc.

Windows is a trademark of Microsoft Corporation.

* These are U.S. prices.



ICLUI handles on the behalf of `IListBox`. Virtual selection and enter functions are defined within the `ISelectHandler` class. Also, virtual focus gain and loss functions are defined in `IFocusHandler`. None of these functions perform any processing, as process definition is totally in the hands of the application developer, through functional overrides, to supply the desired

behavior. However, both handlers are tightly coupled to the Presentation Manager control suite through use of the predefined window classes. Therefore, we must write our own handler since both handlers process events based on the `WC_LISTBOX` window class.

`ListBox32Handler` is designed to spy out all of the `WM_CONTROL` notification events of interest and provide a corresponding virtual function. Four functions implemented were mentioned in the preceding paragraph. We have defined two more, `checked`, which supports the check box extension, and `scroll`, which supports scrolling. Whenever a check box is checked or unchecked, the `checked` function is called, and whenever scrolling occurs, `scroll` is called. Again, by default these functions do not do

anything, as you will need to derive from `ListBox32Handler` and override the specific functions to obtain the desired effect. Since integration with ICLUI is one of our goals, we have designed `ListBox32Handler` to give it the same look and feel as the predefined handlers.

ONE SIZE FITS ALL

ICLUI canvas classes have a special requirement that it places on its child controls: knowledge concerning the minimum size of the controls. The minimum size can be communicated to the canvas classes in one of two ways: overriding the protected virtual member function, `calcMinimumSize`, or explicitly calling the function, `setMinimumSize`. The `calcMinimumSize` function returns an `ISize` object, which is, by default, `ISize(100,100)`.

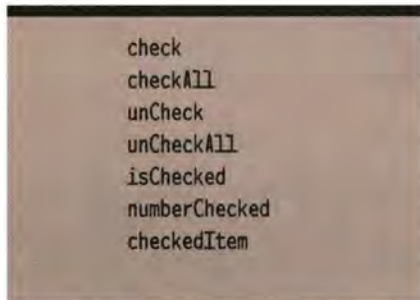


Figure 3. Check-state operation functions

Read and Write dBASE files
from C, C++, and now REXX!

dbfLIB & dbfREXX

dbfREXX allows reading and writing dBASE files from your OS/2 REXX programs!

dbfLIB includes support for DOS, Windows, OS/2 and Windows NT all in one package!



(713) 537-0318

Visa and MasterCard accepted.



dSOFT Development Inc.
4710 Innsbruck Drive
Houston, TX 77066

ASIAN OS/2 DEVELOPER SUPPORT



Get help porting OS/2 applications for Japanese, Chinese, or Korean.

Experienced guidance from the leading double byte character set OS/2 developer. Training or consulting on either an hourly or long term basis.



MicroBurst, Inc.

9035 Shady Grove Court
Gaithersburg, MD 20877

(301) 330-2995

Fax: (301) 330-8609

CompuServe: 70334.3616

Internet: 70334.3616 @
COMPUSE.COM.

Circle Reader Service Number 9

Circle Reader Service Number 10

Additionally, it will not be called if the size has been set via the `setMinimumSize` function.

Although the `IListBox` class does not override `calcMinimumSize`, we would like to make the `ListBox32` class as robust as possible. Therefore, we will override it and supply the minimum required width based upon the longest item in the list box and the current font, if the horizontal scrolling style is *not* used. We will use the default for the minimum height.

MEMBERSHIP INCLUDED

We have added several new member functions to support our check box and sound extensions in `ListBox32`. We would like to concentrate on those functions that were implemented to support the check-state operations and compare and

contrast them to the `IListBox` selection-state operation functions. The check-state operation functions implement support for the extended messages, `LMX_QUERYCHECK` and `LMX_SETCHECK`.

The check-state operation functions are listed in Figure 3. All have been designed to closely parallel the `IListBox` member functions for the selection-state operations, lest we forget our premise of a snug fit with the ICLUI hierarchy. However, we have defined our functions to be virtual. Why? To facilitate subclassing of `ListBox32` and allow an application developer to replace or enhance a function to achieve the desired behavior. Functions in `IListBox` are not virtual; altering behavior will require extra work as well as deviance from standard C++ design. This is

a design issue that will be addressed in a future release of ICLUI.

CONCLUSION

The article text is based on our efforts to design and implement the various classes that wrapper the 32-bit replacement list box. The design and techniques we discuss are applicable to wrapping other replacement controls. You can obtain the source code from the locations listed at the end of this article. In addition to the source for the classes, we have provided a driver application. In the next issue, we will enable text editing in the replacement list box.

We would like to thank Bill Law and Hiroshi Tsuji for answering a lot of our questions and giving us a sneak preview of the book,

SOM

The Object Factory—IDL (Version 2.0)

The first OS/2 SOM tool continues to expand SOM development with the introduction of version 2.0. This feature-packed upgrade forges new ground in state-of-the-art OOUI interfaces, allowing the fastest possible SOM development. The Object Factory Version 2.0 is \$250.

- Full-functioning SOM class browser
- Framework filters & Global class searching
- Complete IDL support
- Supports multiple inheritance
- Includes all SOM 2.0 classes and methods
- User-customizable class development environment
- Drag-drop backups & restorations

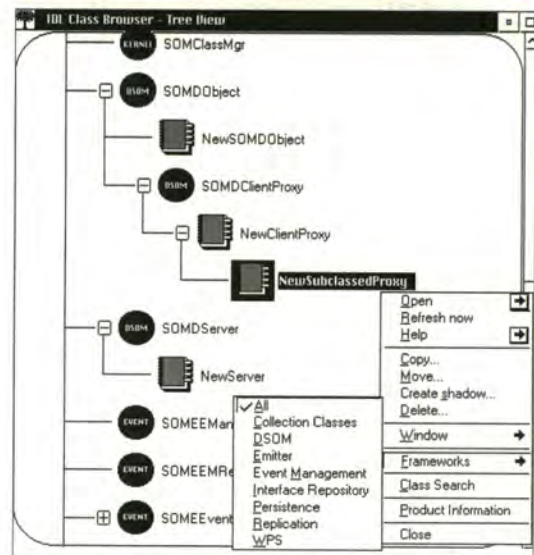
Also available from Synaptec:

The SOM Class Administrator — a complete runtime SOM administration utility presented in a heirarchical tree browser with the following features: Registers & Deregisters Classes, Creates & Deletes Instances, Optimizes System Performance and Handles DLL Version Control & Upgrades. The SOM Class Administrator is \$79.99.

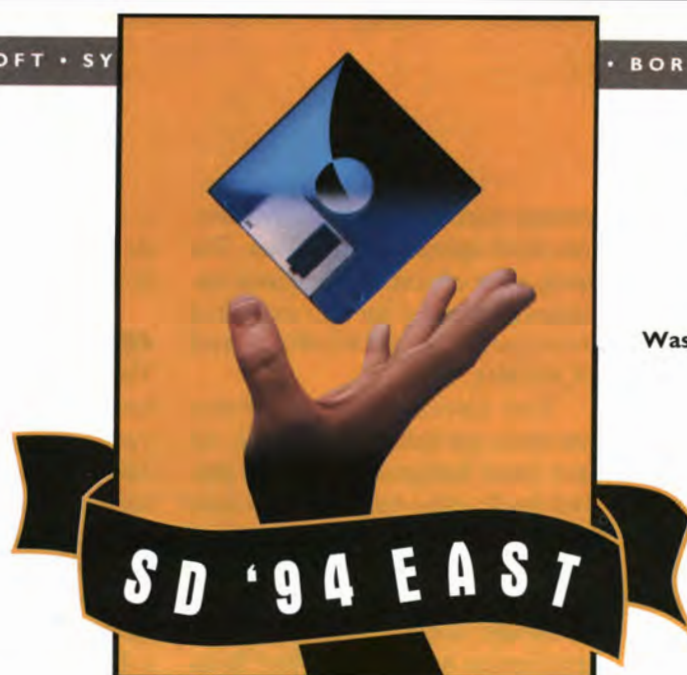
You never knew what was inside your machine! Can be used stand alone, or as an integrated component of The Object Factory.

Call (404)942-8699 for ordering information.

5085 Forest View Trail Douglasville Georgia 30135 (404)942-8699



SYNAPTEC, INC.



Conferences,
October 3 - 7, 1994
Exhibition,
October 4 - 6, 1994

Washington Convention Center
Washington D.C.

Come See the Software and Application Development Industries Come Alive!

"The best speakers, the best demonstrations, and the best exhibition!" Terry McIntosh, Analyst - Chevron Corp.

SD '94 East delivers development solutions across the spectrum of tools from C++ to Powerbuilder.

- Client/Server Tools
- Object-Oriented Tools
- Rapid Application Development Tools
- Database Design and Modeling Tools
- CASE Tools
- GUI Front Ends
- Languages
- Configuration Management Tools
- Testing and Debugging Tools
- Porting Tools
- Multimedia Tools

SD '94 East provides powerful, real-world, real-time education in 4 jam-packed conferences.

SD was the first and continues to be best at bringing you practical, unbiased guidance in over 200 classes across 17 tracks.



Join us at SD '94 East as the industry comes alive...

...on the show floor, in the conference sessions and in a series of special events with key industry players.

- 4 Plenary sessions include:
Mitchell Kertzman, CEO, Powersoft
Christine Comaford, *PC Week*
Vaughan Merlyn, Partner, Ernst & Young
- Hours of free technical sessions by Microsoft, Powersoft, Intersolv and others
- The C++ Programming Superbowl



Tools and Techniques for Software and Application Developers

Call for a Brochure!

1-800-441-8826
415-905-2784
Internet: sd94east@mfi.com
FAX: 415-905-2222

OS/2 V2 C++ Class Library: Power GUI Programming with C Set++ (forthcoming from Van Nostrand Reinhold), which they and three other members of the ICLUI team have authored.

Mark Benge, IBM Software Solutions, Cary, N.C., is a staff programmer who joined IBM in 1989 and has worked on various CUA '91 controls for OS/2 2.x. He works in IBM user interface class library development, where he was involved in the implementation of C++ classes for drag and drop in C Set++ 2.1. Benge has a

B.S. in computer science from Western Carolina University. He can be reached on CompuServe at 73532,2063 and on Internet at banzai@vnet.ibm.com.

Matt Smith, Prominare Inc., Toronto, Ont., Canada, is lead architect for the Prominare

Development System, an OS/2 2.x advanced GUI development environment. He has been actively involved with OS/2 since 1988 and cofounded Prominare in 1990. Smith has a degree in architecture from the University of Waterloo. He can be reached on Internet at prominar.io.org.



REFERENCES

C/C++ Tools, User Interface Class Library User's Guide (IBM order number S829-3743).

C/C++ Tools, User Interface Class Library Reference (IBM order number S819-1179).

C/C++ Tools, Programming Guide (IBM order number S61G-1181).

Welcome to C Set ++ Flyer (IBM order number S61G1670).

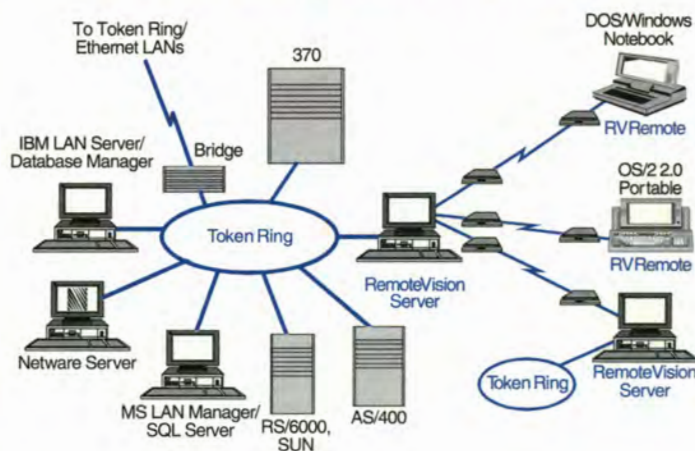
You can download the source code and driver application, titled LSTBX3.EXE, from these electronic sources:

CompuServe's OS2DF2 forum (OS/2 Developer Mag section).

File Area 11 on the IBM PCC BBS, (919) 517-0001.

OS/2 Tools section on the OS/2 BBS for IBM TALKLink customers.

RemoteVision™ MultiPort™ Servers. The Proven Solution For Remote Token Ring Networking.



- Extend LANs transparently over asynchronous modems to remote workstations and workgroups.
- Run DOS, Windows™, & OS/2™ apps written to multiple protocol stacks on remote nodes.
- Operate remote computers as full-function LAN nodes.
- Connect to remote offices with LAN-to-LAN dial-up networking.
- Add remote connectivity services using existing LAN machines.
- Access remote PC servers and hosts without additional host gateways.
- Up to 64 ports on a non-dedicated server

RemoteVision makes it easy and affordable for DOS, Windows and OS/2 users in remote locations to connect to Token Ring or other LANs using dial-up modems. Response time remains fast, applications don't need changing, and users won't need retraining. So you can turn your remote machines into full-function workstations. With RemoteVision. To immediately find out more, or to take advantage of our "Try Now, Pay Later" 30-day money-back guaranteed trial offer, call today.



1265 Montecito Ave., Ste. 101, Mountain View, CA 94043
Tel: (415) 965-8607, Fax info: (415) 965-8607, (then press 2)
CompuServe 71612, 1253. Trademarks are from their respective companies. ©1994 Token Technology, Inc. 001-1

Circle Reader Service Number 12



Corporate Study

Even in a world where they had lost some control of their data, the people at Parkway Insurance found OS/2 still had ways to provide what they needed. **By BRIAN PROFFIT**

Life After Outsourcing



Brian Proffit

When Parkway Insurance shifted its data processing to an outside company, the MIS department found a lot of local development was still needed.

As corporations look for new ways to trim their overhead, many are deciding to hire outside companies to handle a large part of their data processing needs. Internal MIS organizations find their staff decimated while demands from users continue to pour in. Harried data processing personnel find themselves in the middle of several enclosing jaws.

The outsourcing company is usually focused completely on the mainframe processing requirements. Prices for local workstation development projects can be steep, and even then the project is not likely to be high on the vendor's priority list. In many cases, their developers' experience level in the workstation environment is not ideal.

But the vendors hold the trump card—they control the data. Client/server processing is really the only answer. Parkway Insurance required simultaneous access to (but not control over) four host databases and tools that could provide quick development of programs to automate these processes. It found an OS/2 network to be the only acceptable solution.

HOST PROCESSING FRAGMENTED AND SLOW

Parkway is a subsidiary of Fireman's Fund of New Jersey, and the company focuses entirely on personal automobile insurance for that state. Parkway takes customer service seriously, with a group of operators on 800 numbers to answer customer questions.

When they were entirely host-based, those operators faced a rather daunting challenge. The information is spread over four separate databases, so customers had to wait while operators queried each database for bits of the required data. Since each database is keyed on a policy number, operators had to reenter that number four times—with four opportunities to wait for a host response to tell them they'd entered the number incorrectly. Worse still, the operators, in an effort to save time, sometimes skipped the step of writing the policy number down. Unfortunately, they frequently ended up having to ask the customers to repeat the number. This is the sort of problem that led Parkway to a client/server solution.

3270S OUT, VALUEPOINTS IN

Brian Howson, MIS programmer and analyst, reports that nearly 60 new workstations have been rolled in to replace the old 3270 terminals. They



standardized on IBM Valuepoint systems with 16MB of RAM. Most are running at 800x600x256. Connecting all of these machines is a token ring network based on LAN Server 3.01. The server is a PS/2 model 95 with 32MB of RAM and 1.5GB of disk space. Four Hewlett-Packard printers are also on the network, connected directly via JetDirect token ring interfaces. Key to bringing all the data processing needs together is a 3174 controller with a token ring gateway to connect the network with the mainframe.

CONNECTING TO THE HOST

Because that mainframe connectivity is so important—all the mission-critical applications are still on the host—selection of the 3270 emulation software was a critical decision. "I evaluated Attachmate's Extra and IBM's Communications Manager at the same time," Howson reports. "I found Extra to be significantly easier. For example, we have DLC connections. The only thing I have to tell Extra is that it is DLC and give it the destination address. It makes the connection automatically, and I can get sessions up very quickly."

Of course, any decision this important can't be made on the product alone. If you're gambling the success of your business, you have to feel good about the vendor. Parkway has a customer support agreement with Attachmate that Howson would like to see other vendors emulate.

"Our customer support agreement costs \$1,500 per year for the whole site, which is over 50 licenses," Howson says. "That entitles us to single-point-of-contact support. That's sort of like what IBM is starting for OS/2. We get a report on our usage every month, also like what IBM is starting for OS/2. We also get free upgrades, which is not like what IBM is offering for OS/2."

Howson also praises the flexibility of the agreement, reporting that they "have the ability to switch users' licenses from Windows to OS/2. We can upgrade or change the licenses at will."

Of course, the right to free upgrades is only meaningful if those upgrades are released regularly and provide solutions. "They just shipped an upgrade a couple of weeks ago that fixed a lot of things missing from earlier Extra versions," Howson reports. "They added support for the network gateway and added a keyboard mapper as well as fixed a few things that had been bothering us. The program right now is a Mirrors port, so there are some screen flickers and it's sort of slow. In fact, it slows things like our billing inquiry program down."

"Those things will be much faster when they port it," Howson says hopefully. "Right now they're telling me that they should be fully 32-bit OS/2 by the end of 1994. The product manager also told me that they're going to support Workplace OS/2."

CUSTOMER SERVICE FASTER AND EASIER

After the hardware and software were in place, Howson attacked the host database lookup problems the customer service operators were facing. The first step was acquiring a graphic toolkit from Attachmate for accessing Extra sessions, and the second step was pulling it all together with Watcom's VX-REXX.

Howson says, "Billing Buddy [the new customer billing information inquiry system] is a good example of how an inexpensive tool like VX-REXX can exploit the standard scripting [REXX] capabilities of OS/2 to add functionality to a legacy mainframe application quickly and effectively. From the time I got the graphic toolkit for Extra until the application was deployed was under two weeks. That included gather-

ing heavy end-user input, which led to very good acceptance."

The application takes the policy number (which now only has to be entered once) and performs the host database queries automatically—doing so across three separate mainframe terminal sessions

simultaneously. The results are presented in a Workplace Shell notebook. Howson exploits VX-REXX's ability to create multi-threaded programs for responsiveness. "Even though it takes 30 seconds to get all the information, the interface is active all the time,"

he says. "They can begin looking at and paging through the information as soon as the first results are loaded into the notebook. That lets them concentrate on taking care of the customer on the phone instead of fiddling with the computer. It improves customer service and speeds up calls."

VX-REXX BRINGS IT ALL TOGETHER

Billing Buddy is not just faster, it provides the users with new capabilities. For example, Parkway is using cc:Mail as its internal messaging system. If a customer inquiry generates a work item or information request, the operator can click on a button in the billing system and open a cc:Mail message creation screen. As with Extra, Howson was just as careful in the selection of (and is just as high in the praise of) the development tool to use for these new client/server applications.

"I think Watcom is one of the best companies I've ever dealt with as far as upgrades and support," Howson says. "Watcom VX-REXX is about the best-designed tool I've ever seen. It's very intuitive. After you've used it for a while, you really don't have to look at the manual much because you can guess what they did and where they were going. You're usually right because they did exactly what you would expect. It's very consistent." Howson is anxiously awaiting the new Client/Server Kit from Watcom to eliminate a lot of convoluted SQL statements and ease access to the DB2 database.

Howson found Billing Buddy to be a perfect application for VX-REXX. "I can't even imagine how long it would have taken in C," he says. "It would have been an awful lot longer. And because this is an application that is tied to mainframe performance, writing it in C rather than REXX would have given us virtually no advantage. It

Supercharge Your REXX!

GAMMATECH™

REXX SuperSet/2

Supercharge your REXX with the most complete set of REXX external functions available - GammaTech REXX SuperSet/2.

With over 300 functions from seven DLLs, the GammaTech REXX SuperSet/2 will initiate File and System operations, issue Network commands, execute Video I/O, manipulate Processes and Semaphores, regulate the MacroSpace and perform Math calculations.

Call us today at (405) 947-8080 and you'll see what REXX can do with some extra power!

- Perform EXECIO functions
- Manage LAN Server users by group or domain
- Perform LAN Server permission functions
- Manipulate LAN Server files, sessions and shares
- Interface with NetBIOS, TCP/IP and Communications Manager/2
- Open, close, query and create semaphores
- Start a thread
- Destroy a process or thread
- Copy, move or rename individual or mass files
- Dump variable pools or MacroSpace to a file
- Load, drop or reorganize functions in MacroSpace
- Perform logarithmic and trigonometric calculations
- Query hardware components
- List current system environment variables
- Write character and attribute strings
- Obtain and set cursor type and screen mode



SoftTouch Systems, Inc.
Workstation Division

1300 S Meridian, Suite 600, Oklahoma City, OK 73108
(405) 947-8080 • Fax (405) 632-6537

©GammaTech, Inc. 1991-94 All Rights Reserved

Circle Reader Service Number 13

wouldn't run any faster, at least not appreciably. And the whole program is less than 1,300 lines of code in REXX."

IF YOU CAN'T SQL IT, SCRAPE IT

Much of the local processing Parkway does is based on the DB2/2 server on the network. Since the data is owned by the host, the company often creates temporary databases on the LAN for immediate needs, knowing that these will need to be recreated nearly continuously. VX-REXX Client/Server will help with some host database queries, but the users are also forced to do some "screen scraping."

This technique uses existing host programs to bring the desired information to the screen. A workstation program then extracts it from the display for reuse. It's a little less elegant, perhaps, but it can be a lot easier and less expensive than contracting with the outside data processing vendor to create new host code. Parkway is using this technique to help streamline its forms processing.

Previously, most of the company's forms were still being filled out by hand and taken to a clerk for typing. Parkway Automated Forms is querying the host for relatively stable information like insurance agents' addresses and scraping information from mainframe screens for data that cannot be effectively replicated. Howson estimates that the time and labor required to generate a form will be cut to one-sixth of the prior effort.

WHAT MORE COULD YOU POSSIBLY NEED?

For reference material on OS/2 development, Howson relies on IBM's OS/2 Developer Connection CD-ROM. "It has helped me a lot as a source of information. One of the best parts is that the on-line books are on there also. I think that

on-line books being included in all of the individual products is pretty much going to die and be replaced by collections like this. The documentation on the CD is really good, and some of the examples there are helpful too. Basically, I use every resource I can get."

Parkway has a CD-ROM drive on its server, but the single drive can be a bit of a bottleneck for the developers. "It [the single drive] causes some contention. We also use the Westlaw database on CD. It keeps us up to date on New Jersey statutes. This is important because



Smart-Lock

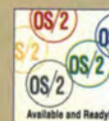
- a comprehensive security and configuration concept
- for host-connected OS/2 workstations
- based on Smart-Cards
- featuring the Chip-Controlled OS/2 Desktop



Smart-Lock is a security concept that controls the access to OS/2 systems by employing Smart-Card technology and a SOM-based restricted workplace shell.

Smart-Lock configures the OS/2 desktop according to the information on the E2PROM chip. Features include user specific boot protection and diskette access, single logon to LAN and host as well as the possibility to encrypt data and faxes.

Smart-Lock is CID enabled and allows easy installation and maintenance in large networks.



Distribution, installation and support in the U.S.A. and Canada through:



International, Inc.

621 NW 53rd Street, Suite 240
Boca Raton, FL 33487 USA
1-800-4SE-INTL or (407) 241-3428
Fax: (407) 278-3919

Circle Reader Service Number 14

we're in a very litigious state, and we have people quoting statutes at us all day in terms of personal auto insurance. So we have that on line nearly all the time." Howson is considering a CD jukebox for the network to allow access to several CDs simultaneously.

EXCLUSIVITY BRING REWARDS

Parkway has standardized on Lotus SmartSuite for its office automation functions—primarily because it is the only real OS/2 suite available, and the benefits of standardization are large. "I still have some users stuck on the DOS versions of 1-2-3 and WordPerfect," Howson says. "We're trying to move people to Ami Pro. I looked at WordPerfect 5.2 for OS/2, and it was really bad. The 5.2A update was a lot better, but by then I had pretty much given up on WordPerfect, and they had given up on us too, I think."

Howson hopes that Lotus Development Corp. doesn't become complacent without compe-

tition in the OS/2 market. "We just got our updated versions of 1-2-3 and Freelance Graphics, but there wasn't much there. Installation is improved—it's faster, more friendly, and prettier. I had to make a lot of noise just to find out about it too. When I was trying to get this upgrade, I could not get any help. I ended up talking to their PR firm, and I finally got a call from their product manager! It was hard to get information from them."

SHAPES IN THE CRYSTAL BALL

Now that the OS/2 network is in place, Parkway plans to exploit it further by tying the users together into a groupware scheduling system. Right now, IBM's Time and Place/2 is the leading contender for this function.

Howson also plans to improve communications with Parkway's vendors by establishing an Internet gateway with the company's cc:Mail system. "We plan on setting up an Internet connection through PSI (an Internet provider),

and we're going to set up the uucp link so we can send mail to anyone on the Internet. With PSI, we can get unlimited Internet mail for the entire office for \$50 per month. And we can use cc:Link to connect the Internet mail to cc:Mail."

In this gateway, though, Howson finds another chink in the Lotus armor. Despite the company's statements of support for OS/2, it appears that not all products are included. Not only is there no OS/2 version of cc:Link, but Howson reports that he has been unable to coerce it into working in a DOS session under OS/2. For its part, Lotus has simply told him that it isn't supported under OS/2, so Howson is being forced to use a dedicated DOS machine for the Internet gateway.

SO WHY OS/2?

Accessing multiple host sessions and a LAN database simultaneously led Parkway to the conclusion that there was no reasonable alternative to OS/2. "I saw after the release of version 2.0 and particularly 2.1 that OS/2 can do the job that other vendors are only now promising," Howson observes. "I need to balance an environment between richness of function and the amount of support it requires. I have found that with OS/2, I can give the users all of the function they need, and I can support the whole network and still run reports and develop applications."

Brian Proffit was part of IBM Corp.'s OS/2 development team before leaving to become director of PC Week Corporate Labs. Proffit is currently president of Visionary Research, an independent consulting firm. He also is the author of two books on OS/2. Proffit is a contributing editor for OS/2 Magazine and has written for PC Week, Dr. Dobb's Journal, and Programmer's Paradise.

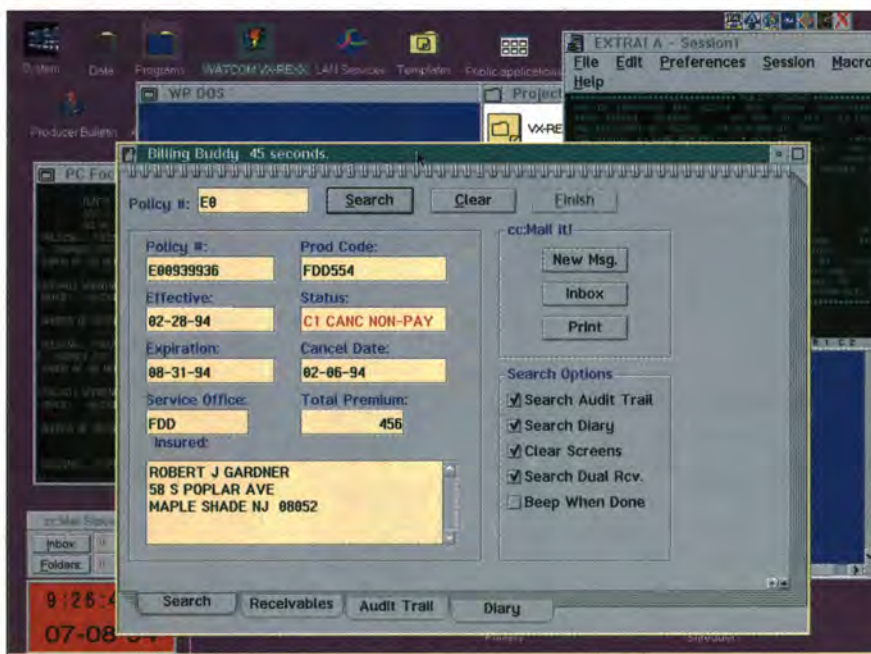


Figure 1. Billing Buddy pulls multiple host databases into an OS/2 interface

LEARN OBJECTS FAST. DEVELOP 10x FASTER.

Want to know the fastest way to learn objects – even for C++ developers? The experts recommend Smalltalk/V, the technology that's 100% pure objects.

FASTER, EASIER LEARNING.

With C++, you have to accomplish two huge tasks: learning object class libraries, and learning new C++ language syntax. Plus C++ is a hybrid of C with added object extensions. So odds are, you'll find yourself constantly falling back on familiar procedural methods and losing the benefit of objects.

Smalltalk/V has a simpler, more approachable language that lets you focus on learning objects instead of a new syntax. That's why thousands of professional programmers have found Smalltalk/V to be the fastest, most efficient way to learn the object-oriented paradigm. In fact,

BusinessWeek

Smalltalk/V to the rescue. Simply by enabling large programs to be built from pretested software objects, it can provide tenfold leaps in programmer productivity and software quality. Perhaps more important, Smalltalk programs deal in terms programmers can understand.

C++ will make more sense once you learn OOP with Smalltalk/V. But we think you'll want to continue using Smalltalk/V for your application development.

DEVELOP 10x FASTER.

Greg Voss of *Windows Tech Journal* says: "It's not an exaggeration to say that most applications of significant size can be built ten times faster in Smalltalk/V than in C or C++." Gen Kiyooka of *Windows Tech Journal* agrees: "Nothing on earth can match the efficiency and productivity of experienced Smalltalk/V programmers. Nothing." And *Business Week* reminds us that some Wall Street firms now get their "computer models of

"You can work through a Smalltalk/V tutorial in a month and gain a better understanding of OOP than you'll get from six painful months with C++. What you learn will make C++ make sense."

— Greg Voss, *OOP Alley*

WINDOWS TECH JOURNAL

brand-new financial instruments done in days, not months."

EASY TO USE.

Here's how Smalltalk/V cuts development time by as much as 90%. •Easy-to-use integrated environment. •Incremental compilation shows immediate results of code changes. •Over 350 well-tested reusable classes built-in and 100's more available. •Class browsers and object inspectors to examine code and isolate errors. •Automatic memory management to eliminate memory related bugs. •Easy access to 3GL languages. •Integrated graphic debugger. •Available on Windows 3.1, Windows NT, OS/2 and Macintosh.

So if you want to learn objects fast – and develop 10x faster – punch this line of code into your telephone: (800) 531-2344 Department 706. We'll send you complete information.



SMALLTALK/V. 100% PURE OBJECTS.

DIGITAL



*This issue's Programming Insider discusses options available for writing objects under OS/2. Using Workplace objects with and without writing SOM, new technologies such as Direct-to-SOM and other approaches such as C++ and user interface class libraries are also explored. **By DAVID REICH***

Objective OS/2 Decisions



David Reich

As you know, objects, object-oriented programming, classes, and libraries of classes are the hot topics in the computer industry right now. Many questions need to be answered when it comes to objects in general, not only when using or writing applications for OS/2, but for any system. We will explore the current implementations of objects on OS/2 here, and you will get some insight on where the use of objects is appropriate and where it may not be as advantageous as you might think.

THE BENEFITS OF OBJECTS

Back in the dark ages of procedural programming, the reinvention of the wheel was commonplace. At that time, reuse was copying chunks of code from one program to another. Eventually, the run-time library was born. The libraries were sets of reusable routines that could be linked into or bound to programs, so functions such as writing lines to the screen or opening files did not have to be rewritten (or copied) every time a new program was written.

The run-time library concept was enhanced with the notion of dynamically linked libraries (DLLs). With the older run-time library, the code for the library routines used by the program

was bound to the executable, so in a multitasking system such as OS/2, there could be many copies of a function (such as, `fopen`) resident in physical memory, each copy in a separate program. DLLs work like their predecessors, but the routines in the DLL can be shared by many programs, so only one copy of a routine need be in memory at any one time.

While all of this was going on, there was an ongoing school of thought where rather than improve on the run-time idea, one would create reusable objects instead of just routines that can be called by programs.

Objects can do work, communicate, and are intuitively more understandable by the average person than the run-time DLL, dynamically or statically linked, function library jargon used by programmers. People work with objects in everyday life; now programmers can too.

Object technology, other than being more intuitively understandable, provides benefits such as data hiding, encapsulation, and, most important, the ability to form the basis for other classes of objects where behaviors can be inherited, modified, and added to.

Object class libraries further isolate programs from the underlying hardware platform, and with the object interface

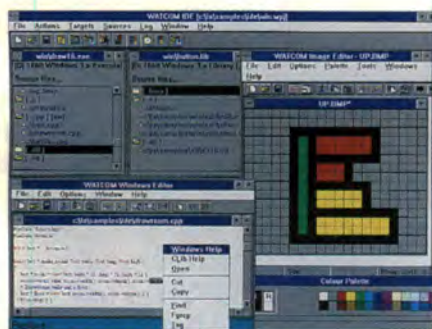
Watcom C/C++ 10.0

ACCELERATE
Your C and C++
Application Development

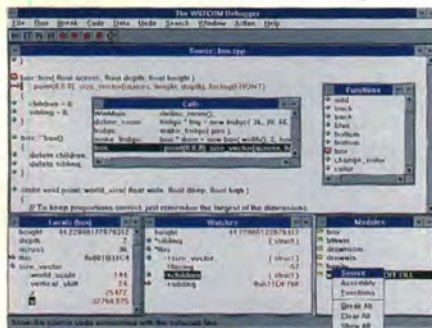
The new Watcom C/C++ 10.0 development system simplifies and accelerates development of high-performance, multi-platform 16- and 32-bit applications. Watcom C/C++ 10.0 delivers productivity and performance, combining our state-of-the-art compiler technology with a new, integrated development environment (IDE) and comprehensive set of tools.

New Integrated Development Environment and Tools

The new IDE is built to simplify the complexities of real-world application development and make it easy to exploit the high-performance, multi-platform power of Watcom C/C++ 10.0. In a single "project" you can build multiple EXEs, DLLs, and LIBs, targeting several different platforms. The IDE simplifies each stage of development from compiling and linking to debugging and performance tuning. The package includes versions of the IDE and tools for all three host platforms (Windows 3.x, OS/2 2.x and Windows NT).



Watcom C/C++ 10.0 includes a source editor with syntax highlighting, a suite of resource editors, testing and monitoring tools for Windows 3.x and NT development.



The advanced multi-platform debugger accelerates the development cycle by increasing the bandwidth between you and your application.

Multiple Platforms in a Single Package

Watcom C/C++ 10.0 supports development of applications targeting an incredible array of platforms: DOS, Windows 3.x, OS/2 1.x, 32-bit DOS (includes royalty-free DOS extender), OS/2 2.x, Windows NT, Win32s, 32-bit Windows 3.x and Novell NLMs. To maximize the potential on individual platforms, Watcom C/C++ 10.0 extends the capabilities of the core, multi-platform toolset with platform-specific tools, SDKs and libraries. This extensive support is amplified by the cross-platform capabilities of the IDE and tools, which enable building applications for a wide range of target environments from any of the host systems.

The Best Optimization Technology

Watcom C/C++ 10.0 combines both 16- and 32-bit compilers in a single package, providing you with the industry-leading optimizing compiler team. PC Magazine tested performance of industry standard C and C++ compilers and said: "the fastest executables created during testing came from Watcom C/C++³², Version 9.5, while the 16-bit version of the same compiler produced the smallest executables³¹". Now, with Watcom C/C++ 10.0, this competitive advantage is delivered with our easy-to-use development environment and tools.

Watcom C/C++ 10.0 delivers all this in a single package!

- New integrated development environment hosted on Windows, OS/2 and Windows NT
- Comprehensive suite of multi-platform development tools including debugger, browser, profiler and more
- Professional source editor, resource editors, testing and monitoring tools hosted on Windows and Windows NT
- Target Platforms include:
 - 16-bit:** DOS • Windows 3.x • OS/2 1.x
 - 32-bit:** Extended DOS • Windows NT
 - Win32s • OS/2 2.x • 32-bit Windows 3.x • Novell NLM
 - AutoCAD ADS/ADI
- Both 16-bit and 32-bit compilers for C and C++, the industry's best code optimizer, faster compile times with pre-compiled headers, C++ supports templates, exception handling and the Microsoft Foundation Class library (MFC)
- Licensed components from:
 - Microsoft Windows 3.1 SDK
 - Microsoft Windows NT SDK
 - Novell NLM SDK v4.0
 - IBM OS/2 Toolkit v2.1
 - Microsoft MFC Class library
- Includes Rational System's DOS/4GW 32-bit DOS extender with royalty-free distribution
- Significantly expanded and revised on-line documentation
- And more!

Suggested Retail Price:

Watcom C/C++ 10.0 CD-ROM Edition
(CD-ROM with on-line documentation) **\$350***

Watcom C/C++ 10.0
(CD-ROM with printed documentation) **\$450***

Upgrades:
(for owners of Watcom C/C++³² or Watcom C/C++ v9.5)

Watcom C/C++ 10.0
CD-ROM Upgrade Edition **\$149***

LIMITED TIME \$199*
INTRODUCTORY SPECIAL
Watcom C/C++ 10.0 CD-ROM Edition

1-800-265-4555

Watcom
A Powersoft Company

Watcom International 415 Phillip Street, Waterloo, Ontario, Canada N2L 3X2 Telephone (519) 886-3700 Fax (519) 747-4971

* Price in US dollars. Does not include freight and taxes where applicable. Authorized dealers may sell for less. \$199 Special Offer is available until October 31, 1994. Watcom and the Lightning Device are trademarks of Watcom International Corp. DOS/4GW is a trademark of Rational Systems Inc. Other trademarks are properties of their respective owners. ©Copyright 1994 Watcom International Corp. PC Magazine, March 29, 1994

Circle Reader Service Number 16



protocol (in general, just telling an object what you want it to do), they reduce the costs in creating portable applications.

OBJECTS AND OS/2

Early versions of OS/2 (version 1.x) were object-based but not ob-

ject-oriented. Some of the things on the screen looked like objects, and to programmers, some things were programmed and acted like objects (for example windows and window procedures), but they still had procedural bases and lacked many of the benefits that a pure object-

oriented implementation provides.

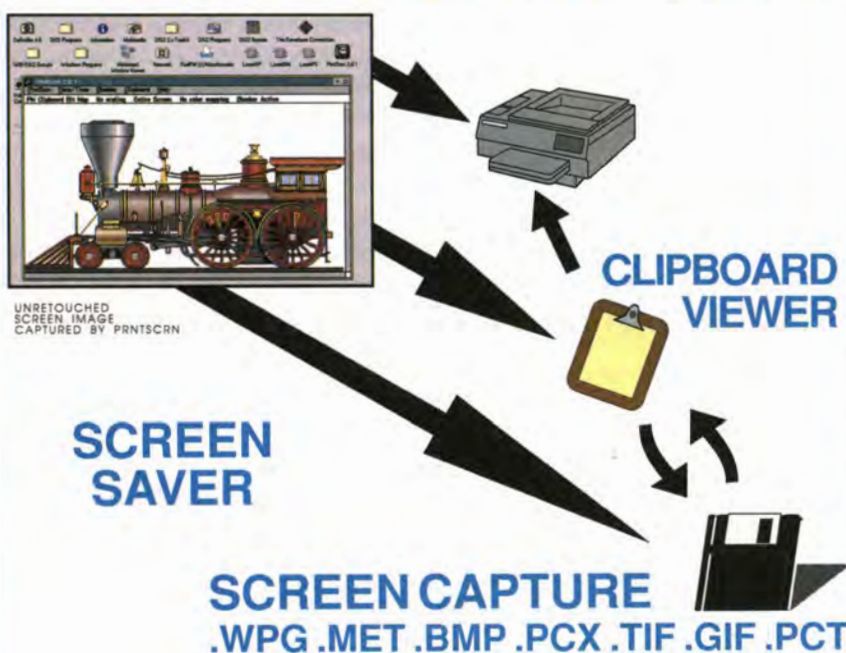
Beginning with version 2.0, OS/2 contains a truly object-oriented user interface, with a real object-oriented engine underneath. This technology is not only available to developers wanting to integrate their applications with or add extensions to the OS/2 Workplace Shell, but it is also there for any other object applications you might think of.

SOM

As you saw in the last two issues, the System Object Model (SOM) is the object technology IBM chose to implement the Workplace Shell. In fact, SOM is the strategic object direction for IBM's software products and the foundation for much of its future technology. SOM is just that—an object technology. As shown in the previous issues with a SOM and Workplace Shell programming tutorial, SOM is essentially a base-class library and a message resolver. All SOM objects are built on top of that base. The SOM run time, or DLL, is the engine that is used by applications. In general, SOM.DLL would have to be shipped with an application. However, because the engine is required by the Workplace Shell, everyone who has OS/2 2.x already has the SOM engine.

SOM also has several flavors. The first release of SOM, and the one shipped with the current version of OS/2, is restricted to a single process. That is, although many processes or applications may use SOM as their object base and engine, if one SOM object wishes to communicate with another, both must be in the same process. That is why any Workplace Shell object you write must become part of the Workplace Shell process for it to use the Workplace Shell object classes and communicate with other Workplace Shell objects.

SINGLE KEY-STROKE SCREEN PRINTING



- Quality Color or Black & White copies of color Desktop Images.
- Copy any portion of any image on the Desktop.
- Screen Saver prevents monitor burn-in.
- Date & Time Display for Time-Stamping images.
- LAN installable. Complete support for LAN attached printers.
- Economical! 4 Utilities for 1 Price! Entity licensing available.
- The most stable and reliable product available.
- Includes both OS/2 2.1/2.0 (32-bit) and OS/2 1.3 (16-bit) versions.

THE LEADER: **PrntScrnm**TM Screen Image Utility for OS/2



MITNOR Software
Phone: (918) 357-1628
Fax: (918) 357-2869

SCREEN IMAGES IN OS/2 DEVELOPER ARTICLES ARE CAPTURED USING PRNTSCRNM, COURTESY OF MITNOR SOFTWARE

Circle Reader Service Number 17

Recall that a Workplace Shell object is a SOM object, but a SOM object is not necessarily a Workplace Shell object.

The SOMOBJECTS toolkit from IBM provides other flavors of SOM that allow objects in different processes to communicate. And there are still other things, such as a SOM Workstation Enabler, that allow objects in different processes on different computers to communicate (often referred to as DSOM) to enable real client/server applications.

WORKPLACE SHELL INTEGRATION

Now that some of the fundamentals are understood, let's look at how to integrate your applications with the Workplace Shell. You already know you can create your own applications or systems using SOM as the engine. Integrating your applications with the Workplace Shell can be done with or without using SOM.

The first question is: How is an application integrated with the Workplace Shell? Some people think they need to write their application as an object. Although that can be done, the idea is to write the application as an operator and the object as the operand.

For example, a document object is a Workplace Shell object that represents a document. In reality, this is (but does not have to be) a data file in the file system. When a user drags a document to a printer object, for example, and drops it there, the object is called or invoked at the appropriate method. In the case of the printer, that method is `wpPrintObject`. If the class that object belongs to handles or overrides that method, the object's class will handle what needs to be done. If not, the method call goes up the ancestry chain to a parent class that does handle that method. This is the advantage of true object-oriented programming.

THE THREE STEPS TO WORKPLACE SHELL INTEGRATION

Workplace Shell integration in your application does not require you to write a new object class using SOM. You of course can, but there are some ways to integrate your application with the shell that do

not require programming to SOM.

Step 1: Basic Drag-n-Drop. The first thing a user is likely to try when confronted with an object-oriented user interface is to take a document object and drag and drop it on a program object. A new object class is not needed to support this.



OnCmd® xBase for OS/2 ... A Winning Combination!

OnCmd harnesses the power and the speed of OS/2, enabling you to develop new applications or convert existing xBase applications, such as those developed in FoxPro®, Clipper®, and dBase®.



Preserving Your xBase Investment

OnCmd can quickly convert your xBase applications to run directly in OS/2's 32 bit Presentation Manager. This minimizes development time and preserves your current xBase investment.



Text applications are automatically converted to a native. Presentation Manager application that is both multi-user and network capable. With simple code changes, you can add buttons, check boxes, and drop-down menus to give your application more functionality.

Applications Development

OnCmd's screen painter, client-server facilities, "event driven" programming, and Dynamic Link Libraries (DLL) all add to the development power that you need.

A Performance Winner

OnCmd is a performance winner! No windows emulation overhead. Disk requirements less than 1 MB.

Installation under 5 minutes. In addition, it typically indexes large databases twice as fast as the competition in half the disk space. For more benchmark details, call or email us.



Client/Server Ready

OnCmd allows you to create a client server environment without the need for a dedicated server system. You can evolve into the world of client/server at your own speed and at a very low cost, allowing you to take advantage of the improved performance and application stability of client/server.

ON-LINE

*Go for the gold.
Order your
copy today!*

ON-LINE DATA

5 Hill Street, P.O. Box 65, Kitchener, Ontario, Canada N2G 3X4
Tel: (519) 579-3930 Fax: (519) 579-2130
CompuServe: 70022,104 Internet: oncmd@onlinedata.com

On-Line Data recognizes the following trademarks: FoxPro (Microsoft Corporation), dBase (Borland International Inc.), Clipper (Computer Associates International Inc.), OnCmd (Online Data).

Circle Reader Service Number 18



When a user drops a document object onto a program object, the shell starts the program and passes the name of the data file the document object represents as a "command line" parameter (that is, the `argc`, `argv` values in the case of a C language program). If your application handles command-line parameters, you already support drag-and-drop object-oriented application launching.

Step 2: Associations. The next step is associations. Wouldn't it be nice if a user could simply open (by double clicking) a document, and it would be presented in the way that person is used to seeing it? Using only Step 1, you cannot provide that. However, using associations, you can provide users this function as well.

Associations are a Workplace Shell feature that associate file types with executable programs or program objects. Some are provided by the operating system (for example, when you click on the README file and it comes up inside the system editor); others can be created by users, and you as a programmer can build associations into your programs.

In general, an association is created between an executable program and a file filter (such as `*.DOC`) or file type (such as "My Document"), which may or may not have file filters underneath. When an object is opened, the shell knows if that object type or if the file extension has an association, the shell will launch the associated application and pass the data file name as a command line parameter as explained in Step 1.

Associations can be created by the user with the settings notebook for a program object.

You as a programmer can relieve the users of your applications from having to set up associations by using an `ASSOCTABLE`. `ASSOCTABLE` is the statement and table you build into the resources of your application. When an executable file is first looked at by the shell (for example, when you create the program object for the shell), the `.EXE` file is examined to see if there is an association table. If so, the shell builds the associations defined in it. This way, you can set up the associations without the user having to do it.

A sample `ASSOCTABLE` is shown in Figure 1. In this example, the name of the association is "My Document", and the file filter is `*.MD`. When any document object that represents a file with the `.MD` extension is opened, an attempt will be made to start the executable program, loading this data file. A custom icon is also in this example. When a folder containing document objects that are `*.MD` files is opened, each document object will have `MYDOC.ICO` as the icon.

Another advantage of using an `ASSOCTABLE` instead of having the user create the association is that when the `ASSOCTABLE` is read by the shell, it creates a template in the templates folder for each document or object type in the `ASSOCTABLE`. Now your users can create new documents by simply "tearing off" a new one from a template.

Step 3: An Object Class. As you can see, you can provide your users a great deal of functionality

without writing your own object class. However, if you want or need to provide more shell interaction than described here (for example, drag-and-drop printing with other than plain text or printer-specific files), you will need to write a Workplace Shell object class.

Writing a Workplace Shell object class today involves writing SOM code, using the Workplace Shell object class hierarchy as a basis, and creating a DLL for your class that will become part of the shell process. This was described in detail with some sample code in "More InSOMnia and Workplace Shell Programming (July/August 1994).

One of the reasons I use the qualifier "today" is that there is work going on by some compiler vendors to provide Direct-to-SOM, which I will explain in a moment.

The net result is that Workplace Shell integration does not require SOM code to be written. Writing SOM or Workplace Shell object classes, however, can ease portability and increase the amount of code reuse you can enjoy.

TOOLS

At this time, SOM tools are, shall we say, less than numerous. More to the point, other than the SOM emitters that come in the OS/2 Developers' Toolkit or the SOM-OBJECTS Toolkit, there really aren't any yet. However, some are in the works.

SOM AND C

SOM itself is really not a programming tool but, as shown previously, more of an engine. To use SOM today, however, the emitters are required to process (or pre-process) the language-neutral SOM source code into source for the target language compiler (most likely C or C++).

```
ASSOCTABLE
BEGIN
    "My Document", "*.MD", AF_DEFAULTOWNER, MYDOC.ICO
END
```

Figure 1. A sample association table.

We're not playing games in Atlanta.

Sprinters, swimmers and pole vaulters need not attend. This world-class event in Atlanta is for software developers, LAN managers, technical coordinators and



systems integrators, who compete on an entirely different level every day. The OS/2® Technical Update '94 is coming to

Atlanta this Fall, and it's where you can get information on IBM Personal Software Products that can help make you and your organization more productive and successful.

During the four-day conference, you can choose from 150 educational sessions conducted by IBM developers and leading industry consultants and educators covering application development, C Set++™ Object programming, communications management, client/server solutions, multimedia,

pen computing, object-oriented technology and more. Special events include "Ask the Experts" Q&A sessions, "Birds of a Feather" informal idea-sharing sessions, and Professional Certification testing. Plus you'll receive pre-release versions of software and invitations to great social events.

Registration is \$800 before August 1, \$895 after, and includes free admission to the NetWorld™+ Interop® '94 Exhibition with over 550 vendor exhibits. A nominal \$200 fee buys "gold pass" access to all NetWorld+ Interop educational sessions. To register or get more information, call 1 800 636-6634. And let's get down /2 business.

**Georgia World Congress Center,
Atlanta, September 11-14.**





The language-neutral code is processed by the SOM emitters and produces C code that is input to your C compiler. The resulting object code is subsequently linked into a DLL that is the executable implementation of your class.

COMPILERS AND C++

Nothing is wrong with choosing C++ to implement objects and object classes in your applications. There is no law saying that SOM is the only thing you can use. In fact, extensive C++ class libraries are now available for OS/2 C++ compilers. Object-oriented programming using these class libraries is a viable alternative to SOM, especially if you are not particularly concerned about porting to SOM later, or if your application platform has no SOM implementation.

However, if you plan on designing your applications for IBM operating systems, you should seriously consider SOM as the object engine, since that is where IBM is heading.

DIRECT-TO-SOM

As you have probably already concluded, a SOM preprocessor or emitter is not the most desirable way to write object code. You have to write the language neutral code, process that, merge it with your C or C++ code, and then compile it. That is what it takes today to use SOM.

Wouldn't it be nice if you could embed code in your C or

C++ files that allows the objects and classes you construct to use the SOM engine and base class implementation? That is where Direct-to-SOM (DTS) comes in.

DTS is being implemented by some compiler makers, so code written in C or C++ can interface directly with the SOM engine, and no preprocessing is needed. Unfortunately, no DTS compilers are commercially available today, but they are in the works. DTS will take the extra preprocessing step out of the equation, and you will not have to learn SOM coding.

OTHER OBJECT LANGUAGES

Although we've only discussed using the more traditional C, C++, and SOM for writing object-oriented code, there are other tools, such as some REXX-based object-oriented programming tools. With these tools, you can take predefined object classes and, using some of their visual drag-and-drop programming models, quickly create REXX applications.

WHETHER OR NOT TO WRITE AN OBJECT

So, should you write an object? If you want to integrate your application with the Workplace Shell, look to see how much integration you want or need. If all you're looking for is simple application launching and document manipulation, you can use associations. If you want more custom behavior in a wider variety of situations, a

SOM or Workplace Shell object class is what you want.

As you can see, there are many choices. Workplace Shell object programming is a very focused topic. For other implementations than Workplace Shell integration, you have choices such as C++, SOM, or even some REXX-based tools. Your object choices should be determined by how portable you need the code to be and what overhead you are willing to live with (to use the object class run-time libraries and so on).

Object-oriented programming is where things are going. Some of the tools available right now are somewhat primitive, such as having to emit C files from SOM rather than having DTS functions available in compilers. But others are full-functioned integrated programming environments. Things are advancing quickly. The best advice I can give is don't waste time using technologies that don't look like they're here for the long run. SOM is. Not only is IBM behind it, but other companies are as well. As operating systems evolve, so will SOM and its tools.

David Reich has been with the IBM OS/2 development team since 1987. He has worked on many parts of the system, supported customers and application developers, and traveled the world giving seminars and teaching OS/2. He is the author of, *Designing OS/2 Applications*, published by John Wiley and Sons. He can be reached on CompuServe at 76711,632 or via Internet at speedracer@vnet.ibm.com.



This article explains how OpenDoc, Taligent, and SOM come together and what they mean to developers. **By CLIFF REEVES, BOB ORFALI, and DAN HARKEY**

A Distributed Object Road Map For Developers

The new generation of multitasking desktop operating systems—such as OS/2—has the potential to radically change the nature of distributed computing. Instead of the classic client/server model, we can envisage the day when hundreds of millions of machines will be both clients and servers. To take advantage of these new forms of distributed computing, information systems will be created by assembling live blobs of intelligence and data in an infinite number of Lego-like arrangements. Today's monolithic applications will be broken down into self-managing components that can play together and roam across networks and operating systems. Distributed objects are our best hope for realizing this potential—they represent the ultimate form of client/server distribution.

Objects are self-managing, intelligent entities that are infinitely malleable. They allow us to distribute intelligence at a very fine-grained level and at the same time can be used to encapsulate existing applications. However, for objects to realize their distributed potential—both on the desktop and across client/server networks—they require an infrastructure. We will use a hardware analogy—the IBM PC—to explain the pieces of this infrastructure and how

they come together. We explain where CORBA, SOMobjects, OpenDoc, and Taligent fit. The PC created an extremely successful industry. The distributed object infrastructure offers a similar opportunity for software developers.

ORB: THE SOFTWARE BUS

The single most important component in the original PC was the hardware bus; it was open and allowed any vendor's adapter to plug in. The object equivalent of an open bus is the Object Request Broker (ORB). Fortunately, this time around, our industry anticipated the need for a software bus and the Object Management Group (OMG)—a consortium of 400 vendors—was founded in 1989 to create distributed object standards before any major products were introduced. The result was the Common Object Request Broker Architecture, or CORBA.¹

CORBA defines how server objects expose their service interfaces to the bus and how clients can call these server objects using static or dynamic API method invocations, as shown in Figure 1. Currently, the CORBA 1.1 software bus supports cross-language and cross-platform object interactions. Commercial examples of CORBA-compliant ORBs include IBM's DSOM, Hewlett-



Cliff Reeves



Bob Orfali



Dan Harkey

Packard's DOMF, and SunSoft's DOE. CORBA 2.0, due late 1994, provides a standard for intergalactic object communications across multivendor ORBs.

THE COMMON OBJECT SERVICES EVERY BUS MUST PROVIDE

One of the painful lessons learned from the PC was that the bus quickly became the single worst bottleneck in the system—it was not designed for high-bandwidth and expansion. Luckily, the object industry did not fall into this trap. OMG divided the bus design activity into the ORB and the common object services that every ORB must provide, as illustrated in Figure 2. These add-on services are object classes that are defined using the CORBA Interface Definition Language (IDL).

So instead of one giant monolithic bus, OMG created a bus with modular add-on object services, each defined by leading industry experts in a software middleware area. Each add-on provides an essential object service for the bus. This is the ultimate in modular bus design. OMG has currently defined standards for four Common Object Services:

- The object lifecycle service defines operations for creating, copying, moving, and deleting objects on the bus. These operations can also handle associations between groups of related objects. For example, if you move a document, the service is smart enough to also move its embedded object parts; this will help us deploy nomadic objects that roam on the bus. The ser-

vice also defines interfaces to factories that can create new objects on different machines.

- The object persistence service allows objects to persist beyond the application that creates the object or the clients that use it. The service is designed for maximum flexibility. It can accommodate a variety of storage services, including ODBMSs, RDBMSs, and document filing systems (like Bento). The idea was to create an open implementation that meets the different storage requirements of objects; it encompasses the needs of large-grained objects (such as documents) as well as fine-grained objects (such as SQL table rows).
- The object naming service allows objects on the bus to locate other objects. This service can use existing network directories such as ISO X.500, OSF's DCE, or SUN's NIS. The service supports naming hierarchies that allow clients to navigate different naming context trees in search of the objects they're looking for.
- The object event service allows objects on the bus to dynamically register or unregister their interest in specific events. An event is an occurrence within an object specified to be of interest to one or more objects. A notification is a message sent to the specific parties informing them that a specific event occurred. Normally, the object generating the event does not have to know who the interested parties are. This is all handled by the Event Service, which creates a loosely coupled communication channel between objects on the ORB that don't know much about each other.

OMG is working on additional ORB-based object services, including transactions, security, concur-

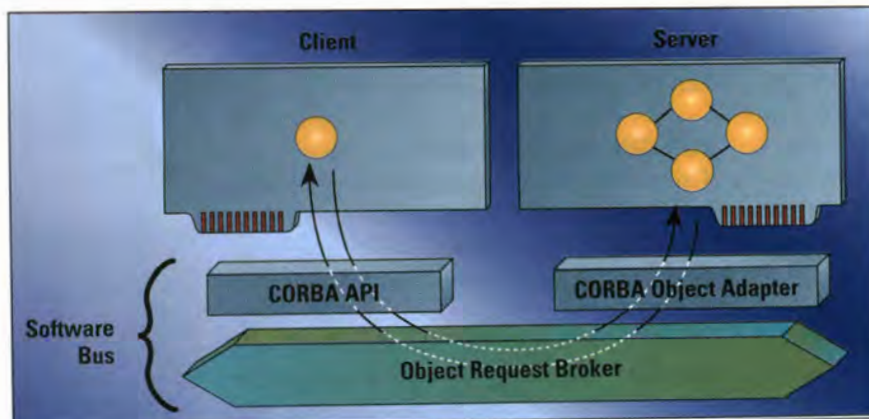


Figure 1. Client/server object interactions on a CORBA ORB

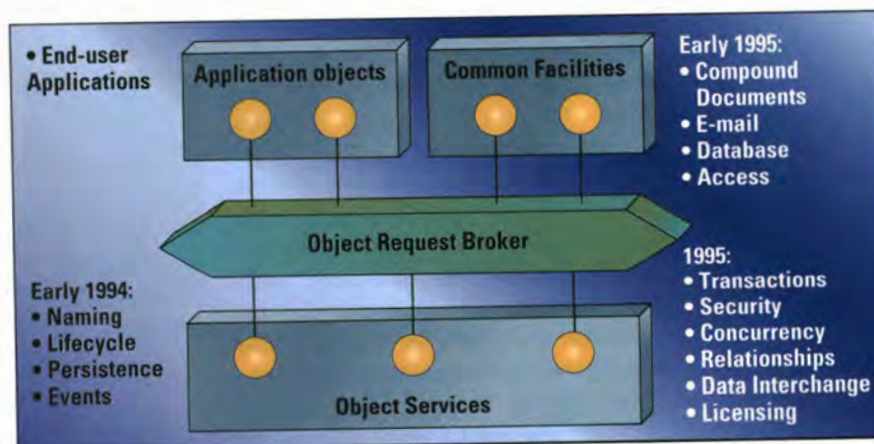


Figure 2. The components of the OMG software bus for objects

rency control, licensing, security, query, and object properties. When they are ready, these services can be added to CORBA-compliant ORBs incrementally. A wide coalition of vendors is using this common object infrastructure to build the next generation of distributed services. CORBA-compliant ORBs—like DSOM—will be able to provide an incremental set of core services for distributed objects. For DSOM, IBM will provide these services first in toolkit form—for example, SOMobjects—and then incorporate them into OS/2 (and other operating systems). This will ensure that the object bus is ubiquitous.

To go back to the PC analogy, we now have a well-engineered software bus and the chassis to house it. In contrast to the PC

hardware bus, our software bus is being built from the ground up to support intergalactic communications (for example, DSOM and DOMF both will run on top of the OSF's DCE). And in contrast to the PC bus, the software bus is an

open industry standard—it is not controlled by a single vendor. But where is the equivalent of the adapter boards that do all the real work on the software bus? This is where object frameworks come into the picture.

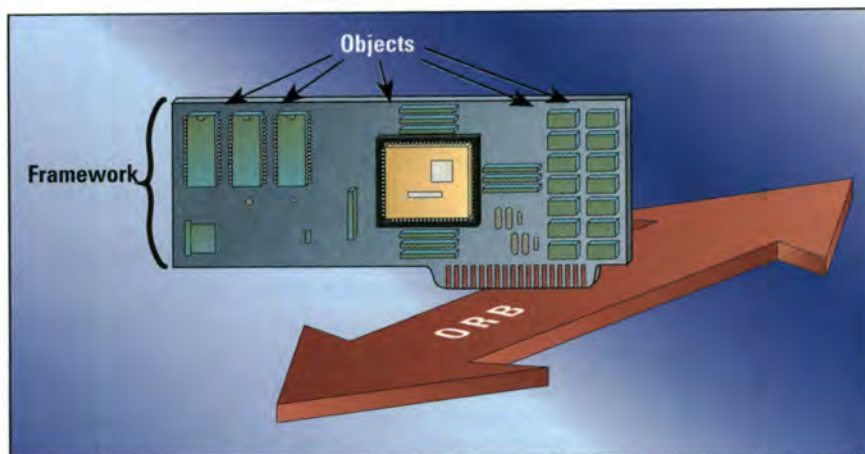


Figure 3. Frameworks: prewired object software boards



BACKUP, RESTORE, MANAGE AND SECURE YOUR WORKPLACE SHELL DESKTOP WITH **DeskMan/2**



EDITORS' CHOICE AWARD
OS/2
MAGAZINE
1994



**NEW!
VERSION 1.5
NOW AVAILABLE**

DeskMan/2™ is the first and only tool designed to manage the Workplace Shell™!

DeskMan/2 for OS/2 is unmatched in its ability to fully exploit the power and versatility of OS/2! Managers can use its security features to protect access to windows or objects. Administrators can easily and quickly standardize a department's desktops for maximum job efficiency. Individual users can quickly and easily migrate icons, objects and desktops between office and home machines. Or use **DeskMan/2** to backup and completely restore a custom desktop.

Incorporating the powerful **VUEMan/2™** virtual desktop & window manager; the potent new WPS snapshot facility; and more, **DeskMan/2** is packed with features!

New & Enhanced Features:

✓ Enhanced for ease of use ✓ Improved performance & functionality ✓ Improved window management	✓ Available object-level security ✓ Complete control over object menus ✓ New easy-to-use installer
--	--

Significant enhancements to every component!

✓ API & GUI for local & remote access ✓ CID enabled & LAD/2 supported ✓ Delete "undeletable" objects ✓ Prevent the deletion of any object ✓ Query any object for settings ✓ Secure individual windows ✓ Secure groups of windows	✓ Change multiple folders or icons all at once -- just drag and drop ✓ Full administrative control of features ✓ Restore, save, migrate any/all objects ✓ Total management of Workplace Shell ✓ And much more ... ✓ Call DevTech for current info!
--	---

DevTech
 Development Technologies, Inc.
 Software Development & Technology Transfer
 308 Springwood Road, Forest Acres, SC 29206-2113 Voice: (803) 790-9230 Fax: (803) 738-0218

© Copyright 1994 Development Technologies, Inc. All rights reserved. DeskMan/2 and VUEMan/2 are trademarks of Development Technologies, Inc. Workplace Shell is a trademark of International Business Machines Corporation. IBM and OS/2 are registered trademarks of International Business Machines Corporation.

FRAMEWORKS: THE SOFTWARE BOARDS

Hardware vendors understood that an open bus was an open invitation to create adapter boards to provide specialized functions—including LAN communications, storage, multimedia, and fax. Whole new industries were created by wiring together individual hardware modules into adapters that plugged into the bus. Frameworks are the software equivalent of hardware boards, as illustrated in Figure 3. Frameworks are a set of software objects that collaborate to provide a special function. Think of them as prewired software classes that do something useful on the bus. They are factory-debugged software subsystems or software boards.

Unlike hardware boards, you can customize a software framework to suit your application needs. You customize a framework by telling it which key events you want to personalize and providing the code that handles those events. The framework then calls your code when that event occurs; your code doesn't call the framework, as shown in Figure 4. Your programs don't have to worry about structure, flow of execution, or calls to system-level API libraries.

If you're an object-oriented programmer, frameworks provide architectural guidance. They remove the need to shop for classes and discover which methods are available, how to call them, and in which order. It's like buying a board instead of individual chips. The services provided by the framework are defined using the CORBA IDL. As illustrated in Figure 5, an application becomes a collection of little puzzle pieces that inherit functions from the framework and call the framework APIs via the ORB.

IBM and Taligent intend to ship CORBA-compliant frame-

works for all types of middleware and desktop functions. You can then modify these frameworks to suit your application needs. But IBM and Taligent won't be alone. Obviously, there is room for a vibrant collection of innovative frameworks on the software bus to come from many vendors.

Table 1 provides a quick summary of the features that distinguish a framework from procedural API programming and object-oriented class libraries.

OPENDOC: CONNECTING TODAY'S DESKTOP TO THE BUS

The software bus is a prerequisite for creating a component-based software industry. However a bus alone is not enough, especially in complex areas such as the desktop,

where there's a need for protocols that allow components to visually coexist, share containers of data, and respond to events (and scripts) in a unified manner. In other words, we need a desktop infrastructure that sits on the bus and defines the rules of engagement by which the visual components can plug-and-play. OpenDoc provides such a standard.² In our bus analogy, OpenDoc is a standard for what goes on inside a desktop software board, as illustrated in Figure 6. To use the hardware analogy, OpenDoc is a specialized desktop adapter with special sockets that let you plug in your parts. You can develop your parts by inheriting function from frameworks, encapsulating existing code, or using OpenDoc part-builder tools. An

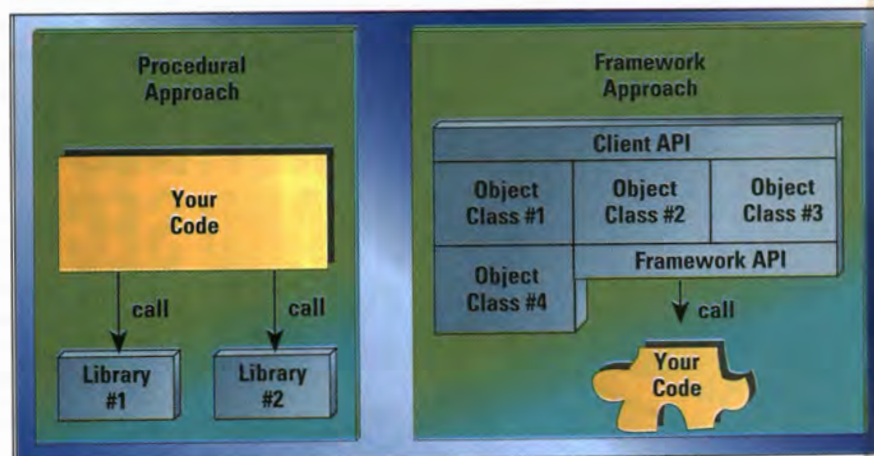


Figure 4. Procedural vs. framework approaches to developing code

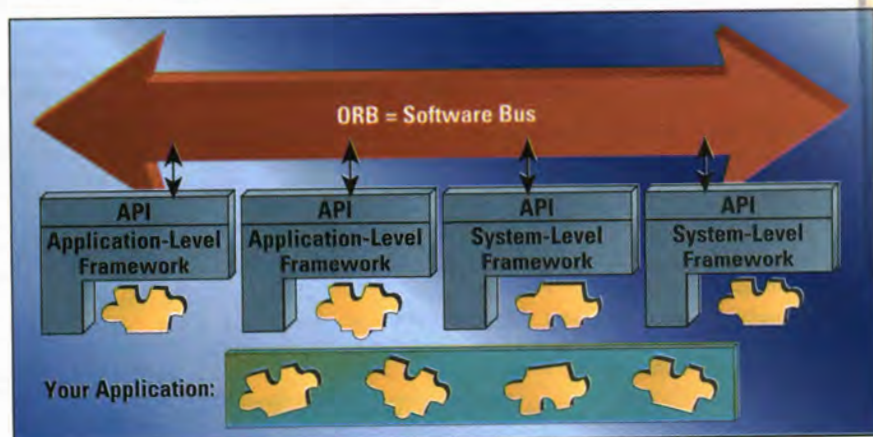


Figure 5. A framework-based application

OpenDoc part is similar to an OLE 2.0 OCX (in fact, OCXs can play inside OpenDoc containers and vice versa).

OpenDoc provides its open sockets for parts by bringing the software bus (or ORB) inside the adapter and defining new services (and APIs) that can be used by software component providers to create visual software parts that plug-and-play into visual and file-based containers. Following are the three new services provided by OpenDoc:

- Bento provides storage protocols for defining rich data types and their boundaries. Bento also defines the protocols by which various parts can coexist in the same file container. Each part can have its own data type and can own a portion of the file container. Bento also allows you to interchange parts across platforms.
- Compound document management defines the visual containment protocols that allow various parts to be embedded and activated from within visual containers. OpenDoc is a form of object-oriented user interface that uses a document-centric paradigm for displaying parts. This paradigm should be very familiar to OS/2 Workplace Shell users. When you need to work on an object, you simply click on it. In fact, parts can be moved from OpenDoc document containers to Workplace Shell containers and vice versa. OpenDoc creates live documents. For example, an OpenDoc document is not a set of "dead bits." Instead, it can be brought to life by a simple click, and its contents can be edited and changed "in place." Meta Group predicts that by 1997, compound documents will become the primary paradigm for capturing information, thus

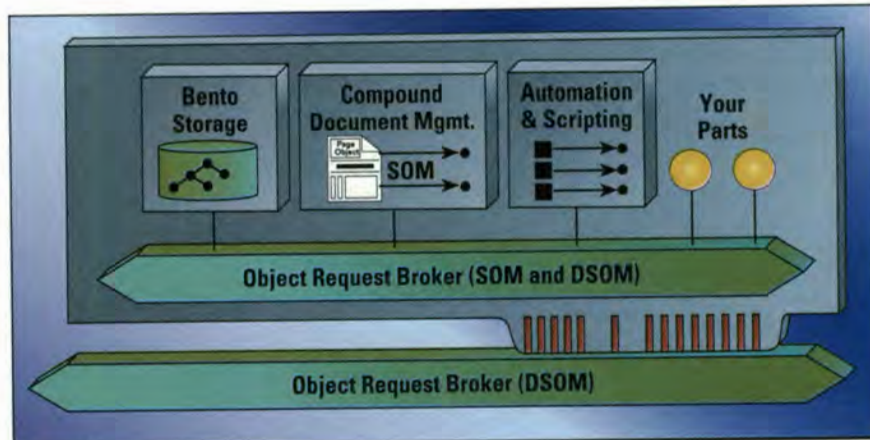


Figure 6. OpenDoc: the desktop software board

challenging the dominance of record-oriented data.

- Open scripting architecture defines interfaces for allowing parts to respond to scripting commands and other external events. Users will be able to

write custom applications by assembling multivendor parts and synchronizing their behavior by writing (or recording) scripts.

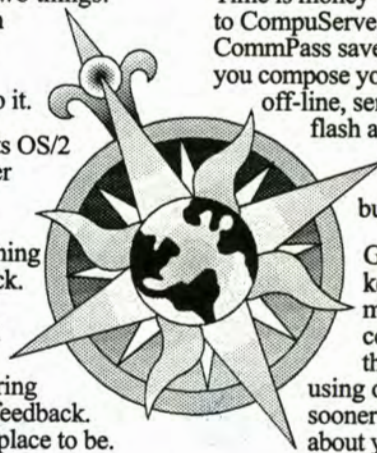
The beauty of OpenDoc is that all the system components and the

GET FAST ANSWERS To Your Development Questions

You need to do two things:

- Go on-line with CompuServe®.
- Use Golden CommPass to do it.

CompuServe hosts OS/2 developer and user forums. IBM developers have responses to anything that's got you stuck. Other OS/2 programmers and enthusiasts are there, too, comparing notes and giving feedback. It's definitely the place to be.



Time is money when you connect to CompuServe, and Golden CommPass saves you both. It lets you compose your questions off-line, send them in a flash and disconnect. It gets your replies while you're busy programming.

Golden CommPass keeps your development schedule on course. The fact is, the sooner you start using our program, the sooner they'll be talking about yours.

Golden CommPass®
OS/2® NAVIGATION SOFTWARE

Creative Systems Programming Corporation
(609) 234-1500 • Fax: (609) 234-1920
Internet: 71511.151@CompuServe.com



Circle Reader Service Number 21



Feature	Frameworks	Object-Oriented Class Libraries	Procedural APIs
Application model	Frameworks are the application. Frameworks handle all the control flow.	You must create the control flow of the application and the glue that ties the different class libraries together.	You must create the control flow of the application and the logic that invokes the APIs. The system knows nothing about your code.
Application structure	Multiple cooperative frameworks.	Single monolithic application consisting of class libraries.	Single monolithic application linked to API libraries.
How services are obtained	Frameworks are the service.	By inheriting function from the class libraries.	By calling API libraries.
How the system is customized	Frameworks call your code. You can subclass parts of frameworks.	By subclassing or creating new classes.	By writing new code and calling additional APIs.
Granularity of control	Medium. You can only subclass parts of frameworks.	High. You can subclass any class.	High. You can write everything from scratch.
Abstraction of services	High. Hides complexity. Automates standard features. You program by exception.	Low. Hides APIs but creates its own layer of complexity—you must determine which methods are available to call and in which order.	Very low. You need to deal with raw APIs and determine the order in which to call them.
How much code do you write?	Very little.	A medium amount.	A lot.
Maintenance costs	Low.	Medium.	High.
Reduced complexity	Yes. You write small pieces of code within multiple frameworks. Frameworks call you only when necessary. Frameworks provide architectural guidance.	No. You must shop for classes and develop the program. You must integrate the different class libraries.	No. You must develop the entire program and understand how the APIs work together.

Table 1. Comparing frameworks, object-oriented class libraries, and procedural APIs (adapted with permission from Client/Server Survival Guide with OS/2)



add-on parts communicate over the same object bus (the DSOM ORB). OpenDoc demonstrates that the same object model—the CORBA ORB—can be used to connect intergalactic objects as well as fine-grained visual objects that reside on a common desktop. It is all very consistent. OpenDoc allows you to repackage monolithic desktop applications into parts that can plug and play together on the same desktop or across the network. You'll need to encapsulate these applications with SOM wrappers and break them into parts and part editors. Taligent frameworks will allow you to go even further and create visual parts that inherit their behavior from prewired frameworks—this is good if you have the luxury of creating new applications.⁴

IF WE BUILD IT WILL THEY COME?

Together SOM, SOMobjects, OpenDoc, and Taligent frameworks provide an object infrastructure plus the basic building blocks that will help developers create parts, components, and vertical frameworks that populate this infrastructure. To use the hardware analogy, writing a part with SOMobjects and DSOM is like developing a hardware chip. Writing an OpenDoc part is like developing a chip that plugs into a prewired board with open sockets—all you do is supply the parts

and they will interplay with other parts. Writing frameworks is like buying a prewired subsystem that you can then simply customize to fit your needs.

Most users will buy either parts they can script together or entire frameworks they can customize for their particular needs. In either case, some form of user customization will become the norm. IBM and Taligent intend to provide the tools that further facilitate this process. The success of the objects industry depends on making our software components as easy to assemble as the new generation of PC plug-and-play busses make it for hardware.

Cliff Reeves is the Director of Object Technology Products, IBM Personal

Software Products Division. Reeves is currently responsible for object-based products in the IBM Personal Software Products Division. This includes responsibility for the IBM/Taligent partnership, SOM, and OpenDoc.

Bob Orfali and Dan Harkey are the authors of the bestselling book *Client/Server Programming with OS/2* (VNR, 1993). Bob and Dan's most recent book is the *Client/Server Survival Guide with OS/2* (VNR, 1994). This 930-page book contains over 150 pages on distributed objects. Bob and Dan have been developing client/server applications and tools for the last eight years. They currently work on the application of distributed object technology. Bob and Dan are affiliated with IBM Personal Software Products Division, Austin, Texas. They work from the San Francisco Bay Area.

REFERENCES

1. Robert Orfali and Dan Harkey, "Client/Server Programming with CORBA Objects," *OS/2 Developer* (September/October 1994).
2. Robert Tycast, "Component Software for the Masses: OpenDoc is Here," *OS/2 Developer* (July/August, 1994).
3. Robert Tycast "OpenDoc Basic Anatomy 101," *OS/2 Developer* (September/October, 1994).
4. Robert Orfali and Dan Harkey, *Client/Server Survival Guide with OS/2* VNR, 1994, IBM publication number SR28-5494. Part 7 provides a 140-page introduction to distributed objects.

The First Twain Enabled Image Scanning & Viewing Utilities for OS/2

ATTENTION

FEATURES:

Fast Image viewing • Easy Screen Capturing • Quick File Conversion • Efficient Color Modification • Powerful Image Manipulation • Real-Time Image Navigation

CURRENTLY SUPPORTS:

Logitech Scanman 256 • Hewlett Packard IIP • Hewlett Packard IIC Series



After The *APPLAUSE* Comes The *REVIEW*

Document Management System

Flexible & Fast Search/Retrieval • Multiple search Fields • Network Support and more...



Solution Technology, Inc. 1101 South Rogers Circle, Bldg 14 Boca Raton, FL 33487
Tel: (407) 241-3210 Fax: (407) 997-6518

Applause and Review are registered trademarks of Solution Technology Inc. All other brand names are trademarks of their respective owners. ©1994 Solution Technology, Inc.

Circle Reader Service Number 33

OS/2 Developer PRODUCTS & SERVICES GUIDE

Katy Computer Systems

OS/2 Developer Special

OS/2 Upgrade edition on CD-ROM \$68

Requires previously installed DOS or OS/2
Price good while supplies last

86 Meramec Valley Plaza Drive • St. Louis, Missouri 63088
314/230-8200

Putting Computers To Work for People

Circle Reader Service Number 23

SMALLTALK Tools

WindowBuilder™ Pro/V is an interactive tool that lets you build polished user interfaces quickly and easily in Smalltalk/V® OS/2. WidgetKits™ provide additional high-level controls, such as spreadsheets, business graphics, table panes, and CUA'91 controls. Full Smalltalk source, no royalties for end-user applications, 90 days free support, and 30-day money-back guarantee. Call for FREE info.

“WindowBuilder Pro/V... the difference between success and failure in Smalltalk/V”

Milan Sremac, President, Medical Software Systems

Objectshare Systems, Inc. v: 408.970.7280 f: 408.970.7282
5 Town & Country Village, Suite 735, San Jose, CA 95128

Circle Reader Service Number 27

ACI Technology Training OS/2 Workshop Schedule

<u>Course</u>	<u>Date</u>	<u>Location</u>
DB2/2 Application Programming	9/26-29	Chicago
OS/2 PM Programming	10/3-7	Chicago New York City
C Concepts for PM Programming	11/1-4	Chicago
OS/2 Programming & Adv. PM	11/14-18	Chicago
OS/2 Introduction	11/17-18	New York City
OS/2 PM Programming	12/12-16	Chicago

ACI is an OS/2 Accredited Training Company
and a member of the IBM Independent Vendor League

***For Sales/Service, please call Robert Anderson at
708/285-7805***

Circle Reader Service Number 22

The easy to use 32 bit OS/2 PM IPF Language Editor...

IPF Editor

- Add help to applications in minutes
- Designed for easy use by programmers and non-programmers
- Generate C and Resource source files to add help to applications automatically
- Import wordprocessing files, including WordPerfect, quickly and accurately
- Preview IPF output without compiling
- Create all hypertext links automatically
- Optional Voice Support Module
- Includes complete on-line help and documentation

IPF Editor: \$150
Voice Support Module: \$50

Orders:

1-800-IPF-7622

Information: (206)428-5025
on compuserve at
70410,2416

Perez Computing Services
4725 Monte Vista Place
Mt Vernon, WA 98273

Circle Reader Service Number 24

Starting today...

There is a whole new way to increase your productivity from using OS/2 desktop applications and Lotus Notes!

SmartLink quickly and easily empowers your applications to use Lotus Notes for Document Management, E-mail, Fax, Paging, OCR, Voice and Video.

Call (617) 424-8302 today for more information.

Circle Reader Service Number 26

TO ADVERTISE IN THIS SECTION CALL: KRISTIN MORGAN @212.626.2498

PRODUCTS & SERVICES GUIDE

OS/2 Developer

Opt-Tech Sort/Merge

High Performance Sort/Merge/Select Utility

Use as a stand-alone utility or call as a Subroutine. Many features including unlimited file size, multiple keys, record selection, duplicate elimination, summing and more!

Call for free brochure Opt-TechData Processing
P.O. Box 678
Available for: Zephyr Cove, NV 89448
OS/2 or Unix \$249 Phone (702) 588-3737
DOS or Windows \$149 Fax (702) 588-7576

Circle Reader Service Number 31

E/R DIAGRAMMER by IRMWARE

Simplicity and Flexibility - Supports Open Architecture, Supports User Definable Standards, On-line Diagram Editing, Cross-Diagram Linking.

E/R DIAGRAMMER - An inexpensive entry into CASE technology that remains methodology independent without losing the precision of binary cardinalities.

We provide Import/Export facilities with source code for easy integration.

FREE PRODUCT DEMOS, BROCHURES OR
CONSULTING INFORMATION:

(602) 730-1510 or fax (602) 413-0399
P.O. Box 51093, Phoenix, AZ 85076-1093

Circle Reader Service Number 29

FAX for OS/2

From the developers of **FaxWorks OS/2** and **PMfax**
Client/Server API and Printer Driver Toolkits
Support for LANs, up to 96 lines per CPU,
and all popular fax hardware

Keller Group Inc.

Voice: (612) 429-7273

Fax: (800) 329-3293

CIS: 72120,3404

Fax: (612) 653-1987

Faxworks is a trademark of SofNet, Inc.

PMfax is a trademark of Keller Group

Circle Reader Service Number 30



REMOTE LAN. NETWORKING

RemoteVision extends LANs transparently over dial-up connections. RemoteVision provides access to a LAN from remote machines or remote LANs. Remote DOS, Windows, and OS/2 machines have a simulated network adapter.

Communications to the LAN is provided by the RemoteVision Server and async modems. Most LAN protocols and applications work, including NetBIOS, Named Pipes, APPC, 802.2, TCP/IP, and IPX. Call (415) 965-8607 and use option 2 for a free fax-back.

Circle Reader Service Number 25



TCP/IP for OS/2®

Supports *all* released versions of OS/2
including 32bit API for OS/2 2.0 and above

**Our customers
say our tcp/ip is the best.
It's faster, more reliable,
and more complete.
Find out for yourself!**

A DAN LANCIANI PRODUCT

For further information contact: Essex Systems, Inc.
One Central Street • Middleton, MA 01949

(508) 750-6200

TCP/2 is a trademark of DLD consulting. Ethernet is a trademark of Xerox Corporation, OS/2 is a registered trademark of International Business Machines Corporation. VT102 is registered trademark of Digital Equipment Corporation. TCP/2 is based in part on work done by the University of California Berkeley.

Available through OS/2 EXPRESS

Circle Reader Service Number 32

TO ADVERTISE IN THIS SECTION CALL: KRISTIN MORGAN @212.626.2498



Object-Oriented Programming

In our previous article on OpenDoc ("Component Software for the Masses: OpenDoc is Here, July/August 1994), we introduced the major technologies contributed by IBM, Apple, and WordPerfect to the Component Integration Laboratories (CILabs) consortium: SOM, BENTO, OSA, and the OpenDoc compound document architecture. This article will focus on OpenDoc's compound document technology.

By ROBERT L. TYCAST

OpenDoc Basic Anatomy 101

In OpenDoc, we stretch the document metaphor in many ways.

From the OpenDoc point of view, everything that a user builds, manipulates, and stores is a document. Diverse elements from simple sound bites to the OS/2 desktop itself can be documents. The sound bite can have a visual trigger, such as a push button or an icon, which causes its contents to be retrieved and played as a dog's bark, a fragment of a song, or a sound effect. At the other end of the spectrum, the desktop is a complex hierarchy of documents within documents. Folders are subdocuments within the desktop, and those folders contain further subdocuments.

In OpenDoc, these subdocuments are referred to as parts that are embedded in some containing document. We will see how these concepts are born of the objects that make up the OpenDoc run time.

PART

A part is the building block of OpenDoc. Every document contains at least one part, and each part owns some real estate in the document called a frame. A part defines the type of data that can exist in its frame as well as the editing conventions that apply. For instance, a structured graphics part would have

data that corresponds to shapes. A text part would have data stored as characters, words, lines, and so on. Graphical objects created by the structured graphics parts might be manipulated by handles. Text might be edited by using cursors and simple delete operations or more complex selections of text displayed as highlighted lines. The point is that each part has complete freedom to define both the data types and the editing conventions.

In a very real sense, the parts correspond to applications in today's world. The first part defined for a document is called the root part, which defines the overall conventions for the document. A simple document will have only one part: the root. So in OpenDoc terms, today's applications are simply documents with only a root part.

More complex documents will of course have more than one part within them. But before we can go on in our discussion we must define some additional terms.

EMBEDDING PARTS

The root part initially owns all the document's real estate in a single frame. But if it is truly OpenDoc compliant, it will be able to create new frames within itself, called embedded frames. (A part is

Don't even think about using a Relational Database System !



Comparing Object Oriented and Relational Design Methodologies

Feature	POET ODBMS	Relational RDBMS
Storing Objects	As Objects	Break Objects into Tables
Database Model	User Application Model	Separate Database Model Required
C++ Integration	Total	Poor
Database Operations	At Object Level	Must Write Code
Productivity	Increased	Reduced
Complex Object Performance	Excellent	Poor

Use the POET Object Database for C++

C++ and class libraries have made the GUI development much easier. After all, it is only logical that more and more developers think in objects. But object orientation shouldn't end at the user interface programming level.

The Problem: Without POET, a C++ programmer must use flat files or a RDBMS to store objects. He has to write code to overcome the mismatch between the application and the database model. This leads to design restrictions, performance penalties and more code to write and maintain.

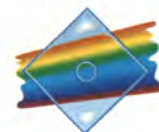
The Solution: POET operates at the object level, it speeds up the development process and provides greater performance. Furthermore, the developer can simply take the object-oriented application design in C++ and map it 1 to 1 into the database. Without any compromise at all!

Full featured database:

POET provides complete support for Encapsulation, Inheritance, Polymorphism and Containers as well as queries, object locking, and transaction handling.

True cross platform support: Complete interoperability makes the development of network enabled applications a breeze. POET supports Macintosh, Power Macintosh, Windows 3.1, Windows NT, Win32s, Windows for Workgroups, Novell Netware (NLM), SUN, AIX, HP-UX, SGI, SCO, OS/2 and NeXTStep.

Call 1-800-950-8845



POET
Software

POET Software Corporation, 4633 Old Ironsides Dr., Suite 110, Santa Clara, CA 95054, Tel.: (408) 970-4640 Fax: (408) 970-4630
United Kingdom (Silicon River) +44 81-316-7777, Germany (POET Software) +49 40-60-99-00, Australia (Microway) +61-3-580-1333, France (LCI) +33-1-34-65-7777
Netherlands (Protools Software) +31 20 645 5023, Sweden (Duke Systems) +46 8 703 2781



not required to embed other parts. It is up to the implementor to decide whether or not to allow embedding. But if a part does not embed other parts, its usefulness is severely limited.) These frames are OpenDoc content objects that contain parts. They are manipulated—moved, resized, and deleted—ac-

cording to the rules of the part, usually the same policies that apply to the part's other content objects.

These embedded frames are special, however. While they are created and manipulated by the containing part, they are given to a new part that governs what is in the embedded frames. This may sound a little confusing at first, but it is the key to understanding what's happening in OpenDoc.

In Figure 1, the root part has two embedded parts—A and B. Part B contains embedded Part C. If the user wants to change the layout of the document by moving either A or B, it is the root part's responsibility to provide the means to select the embedded frames. If Part C must be moved, it would be Part B's responsibility since Part C is embedded in it.

We started off looking at the root part and its capability of creating embedded frames and are now

examining an embedded frame—Part B in Figure 1—that does its own embedding. It turns out that the only thing special about a root part is that it was the first part selected for the document. Otherwise, it operates identically to parts that come later in the hierarchy. You can still embed other parts by creating embedded frames in the parts that require them.

DOCUMENT SHELL

The document shell (DOCSHELL.EXE) provides the document with the basic environment in which the component parts can execute. It provides the process and memory context for the document as well as some common services that all parts will need. For instance, the document shell is responsible for dispatching any messages or events to the correct destination. These might be the result of actions by the user or messages sent between documents and parts of documents.

Access to certain resources must be serialized at a document level for parts to share them. The document shell provides an arbitrator object that enforces a protocol for sharing menus, the mouse and keyboard, modal focus, and so on. Each shareable resource has a focus module associated with it. The mechanism is extensible, and any resource that needs to be shared in a document can use the arbitrator simply by creating and registering an object called a focus module.

Document shells are delivered as part of the OpenDoc implementation on the platform. In most cases, this default "docshell" will be all that is needed since it is the parts in general, and the root part in particular, that actually define the character of the document, not the document shell.

One case in which a software developer might create a document shell is when an existing ap-

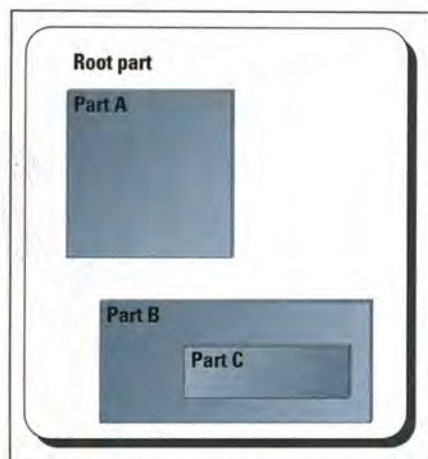


Figure 1. Embedded parts in an OpenDoc document

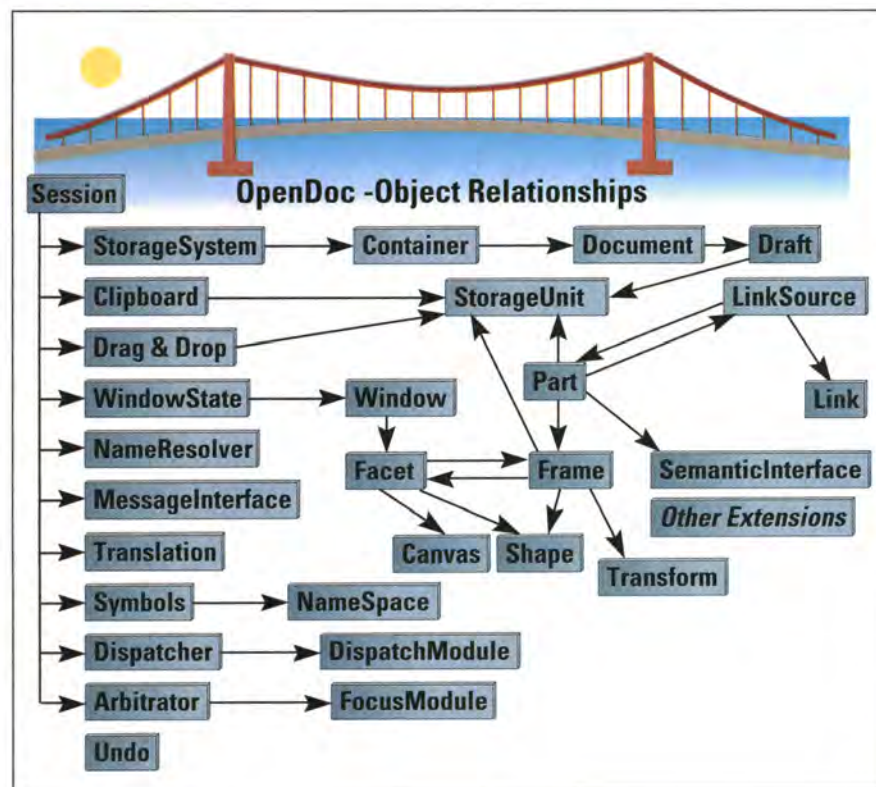


Figure 2. Object hierarchy

plication is being migrated to OpenDoc in an evolutionary manner. As a first step, the application can be modified to contain OpenDoc parts, even though the application itself could not be embedded in an OpenDoc container.

In this case, the application would in effect be converted into a document shell with a hard-wired root part, representing the native data type of the application. At a later date, other steps would be taken to completely decompose the application into component parts. At that point, the application would cease to exist as such, replaced by the set of parts that would now run under the auspices of the default document shell. The specialized document shell produced as an intermediary step would no longer be needed and would be discarded.

PART HANDLERS

The heart of the OpenDoc hierarchy is the part object. It is nothing more than an abstract class with all virtual functions. In the OpenDoc world, it is the part that is delivered to the user, and it is the part handler that implements all the methods of the part object. Part handlers can be divided into two general classes: part editors and part viewers.

Part editors. Part editors are the workhorses of OpenDoc. They provide the functionality to create new part data in an OpenDoc document and to change existing data. Part editors will be marketed and sold as OpenDoc parts individually or possibly as part of a package of complementary parts bundled together for the user's convenience.

Part viewers. Part viewers are scaled-down versions of part editors. They do not contain functions necessary to create new part data. They can only render data within a document. The existence of a part

viewer on a platform, however, ensures that a part's data can be rendered with maximal fidelity on that platform.

The idea is that while part editors have to be purchased to be licensed and used, part viewers

should be made available for little or no cost on all platforms. For any given category of data, text for example, it is likely that each user will have purchased a part editor of his or her preference. When a data file is exchanged among mul-



The secret's out!



- Native OS/2 spreadsheet •
- Objects to the core • REXX scripting •
- Small • Fast • Powerful •

MESA™ 2 
for OS/2®

The spreadsheet that's more than the sum of its parts.

**TO ORDER: CALL
1.800.315.MESA**



17 St. Mary's Court
Boston, MA 02146
phone 1.617.734.6372
fax 1.617.734.1130



Circle Reader Service Number 36



multiple users, it is unlikely they all will have exactly the same preference for a text editor part. But since they all will have a part viewer for the common text formats, they all will be able to view the document with the appropriate part viewer.

OBJECT HIERARCHY

The OpenDoc library includes a set of objects that provide services to the OpenDoc shell and to part handlers. Figure 2 shows the hierarchy. A discussion of the vital objects follows.

Drafts. A draft is a unique version of the document that a user is working on. Each document starts out as a single draft. Each time the user decides to save the document, a new draft is created and the old one archived. So OpenDoc provides

a de facto archiving mechanism transparent to the user. The actual mechanism is extremely efficient since as each draft is saved, only the changes from the previous draft are actually retained.

At any time, the user can decide to retrieve a previous draft and work on it. This sets the stage for future collaboration facilities where different users can work on a document simultaneously, each with their own drafts, without fear of interfering with others.

Storage units. Each draft is made up of one to many storage units. The storage units are used by the part handlers for data. Storage units can be in memory, on the clipboard, as links, or on disk. Since each is an instance of the same class, it is quite easy and convenient for a part handler to put or

take its data from a storage unit interface and let OpenDoc handle issues like making the data persistent, copying to and from the clipboard, or retrieving a link from a remote document. In each case, the interface is always the same storage unit object.

Frames. Parts are embedded in other parts by creating a frame and starting up an instance of that part within it. It is through the frames that are embedded in a containing part that the layout of the document is determined. When a document's draft is saved, the set of frames and their geometry are made persistent.

Facets. Facets represent the areas of a part's frame that are visible during run time. Whenever a part becomes visible, it is given one or more facets, and the part's draw method is called to allow it to render itself. A part may have become visible for many reasons. For instance, it may have just been embedded into a document, an occluding part may have moved, or the user scrolled the page on which the part's frame is located onto the screen.

The facets are ephemeral. During the normal course of editing a document, the facets will come and go. They are not stored with the document data. When an existing document is opened, the containing part will determine which parts are visible based on the stored frame layout, facets will be constructed for all of the visible parts, and the appropriate draw methods are called.

Canvases. Canvas objects are platform-specific objects that represent the context with which the part handler renders. In Presentation Manager, this would be akin to a presentation space or a device context.

There are two basic flavors of canvas: static and dynamic. A dynamic canvas is used to render a

Next issue...
Bonus Distribution at
COMDEX!
Get your sales
message out cost-effectively by
advertising in the
OS/2 Developer
Products & Services Guide
CALL:
Kristin Morgan V/212.626.2498
kmorgan@mfi.com

part on the screen. A static canvas is used for printing. When a part is asked to draw itself, it can query to type of canvas it has, and if it is static, it may decide to render itself differently. Scroll bars are not usually useful on a printed page, so a part handler that normally has scroll bars on the screen will typically omit the scroll bars when printing (that is, it has a static canvas).

Robert L. Tycast, IBM Entry Systems Division, Boca Raton, Fla., is an advisory programmer for the Advanced Presentation Manager Development group. He joined IBM Corp. in 1989 from Digital Equipment Corp., where he served in a

number of software development roles, including providing software support in the U.S., Latin America, and Europe. His project experience over the last 13 years includes X11, AI Technology (LISP and

OPS5 support), and technical workstations (VMS and ULTRIX). Tycast has a B.S. from Massachusetts Institute of Technology and has done graduate work in MIT's computer science department.



REFERENCES

CILabs. Component Integration Laboratories, 688 Fourth Ave., San Francisco, CA 94118, ph: (415) 750-8352 (voice), fax: (415) 757-4829, Internet: jed@cil.orgx

The Developer Connection for OS/2. The OpenDoc Developer's Toolkit program is available by ordering the OS/2 Developer Connection CD-ROM, ph: (800) 633-8266, fax: (303) 330-7655.

Compuserve. There is a section in OS2DF1 for OpenDoc: Section 14, "OpenDoc for OS/2."



You NEED ASSISTANCE!

Who doesn't? Now one tool provides the assistance you need.

Wouldn't you like your applications to work the way you work? The power and complexity of today's applications require compromises in the design, location, and default settings of all features. People often use only a few features frequently, and use each feature in specific ways. PM Assistant quickly executes your favorite features and configures them to work the way you want.

With PM Assistant you use hot-keys and buttons to perform common tasks. These buttons and hot-keys bring applications to the foreground, starting them if necessary. They execute menu commands, initialize or change the settings in dialog boxes, copy data to and from the clipboard and among applications, and arrange your windows on the screen. It's like having a personal assistant doing the tedious elements of your job for you!

Like an able assistant, PM Assistant can perform long, intricate procedures by combining all those types of tasks into powerful compound macros. Imagine typing your closure, printing, saving, and forwarding your document to your manager all with a single key-stroke or the press of a button!

Now imagine performing tasks without even being there. With PM Assistant's task scheduler you can exe-

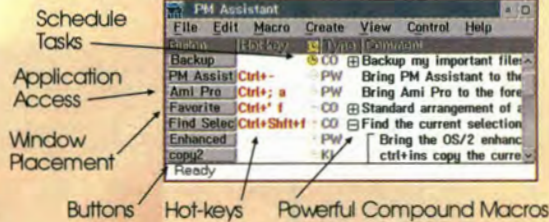
cute macros at any specific or recurring time. You can start programs, backup data, download files, begin long calculations, pop-up reminders... at any time.

The uses for PM Assistant are limitless. You can execute macros indefinitely, and in random order. You can

stress test applications. PM Assistant is ideal for VARS and MIS administrators. You can integrate the power of several applications. You can distribute macro files to other users enabling them to complete complex tasks *without knowing how*. PM Assistant accelerates and simplifies anyone's work.

6 Utilities In One

1. Hot-key access to applications.
2. Window placement.
3. Keyboard macros.
4. Task scheduling.
5. Screen Saver.
6. Password Protection.



Utilis

P.O. Box 367
Redmond, WA 98073
(206) 467-4025

PM Assistant

Call 1-800-745-7757

Demo disk and brochures are available.

\$129.00

Including shipping.
Visa or Mastercard accepted.
90 day satisfaction guarantee.

Circle Reader Service Number 53

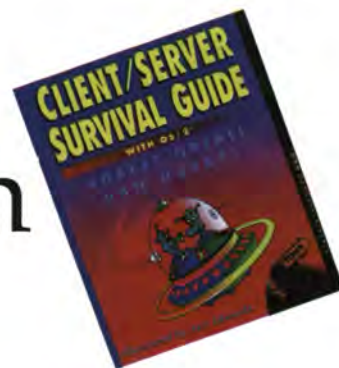


Object-Oriented Programming

This article is excerpted with permission from *Client/Server Survival Guide with OS/2* (VNR, March 1994). 976 pages. ISBN 0-442-01798-7. IBM Publication Number SR28-5494.

By **ROBERT ORFALI** and **DAN HARKEY**

Client/Server Programming with CORBA Objects



Robert Orfali



Dan Harkey

Object Request Brokers—or ORBs, as the natives call them—provide the middleware that establishes the client/server relationships between objects. Using an ORB, a client object can transparently invoke a method on a server object, which can be on the same machine or across a network. The ORB intercepts the call and is responsible for finding an object that can implement the request, pass it the parameters, invoke its method, and return the results, as illustrated in Figure 1. The client does not have to be aware of where the object is located, its programming language, its operating system, or any other system aspects that are not part of an object's interface.

THE STRUCTURE OF A CORBA ORB

Clearly, we need standards for objects to interoperate in heterogeneous client/server environments. Fortunately, this time around the industry anticipated this need, and the Object Management Group (OMG) was founded specifically to create distributed object standards before any major products were introduced—a truly amazing phenomenon.¹ OMG-compliant ORBs are commonly referred to as CORBA, which stands for Common Object Request Broker Architecture.

Figure 2 shows the client and server components of a CORBA ORB. Even though there are many boxes, it is not as complicated as it appears. This section provides the big picture of what these components do. We go into more detail later. The key is to understand that CORBA, like SQL, provides both static and dynamic interfaces to its services. This is because OMG received two strong submissions to its ORB request for proposal: one from HyperDesk and Digital Equipment Corp. that was based on a dynamic API, and one from Sun and Hewlett-Packard that was based on static APIs. OMG told the two groups to come back with a single proposal that combined both features. The result was CORBA 1.1. The “Common” in CORBA stands for this two-API proposal.

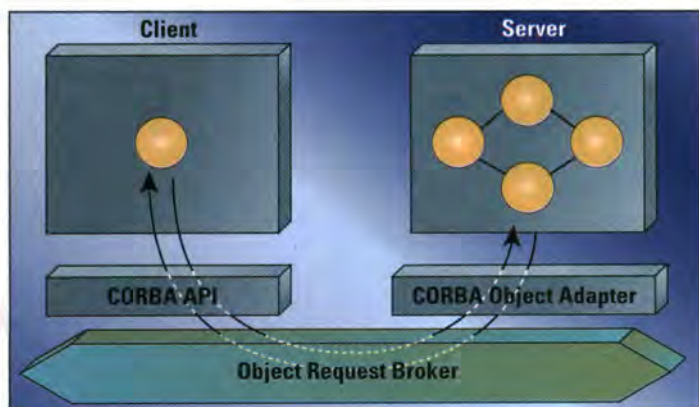


Figure 1. The client/server request using the ORB



Let's first go over the ORB components on the client side:

- The client Interface Definition Language (IDL) stubs provide the static interfaces to object services. These precompiled stubs define how clients invoke corresponding services on the servers. The services are defined using the IDL, and both client and server stubs are generated by the IDL compiler.
- The dynamic invocation APIs let you discover the method to be invoked at run time. CORBA defines standard APIs for obtaining the service definitions, generating the parameters, issuing the remote call, and getting back the results.
- The interface repository APIs allow you to obtain descriptions of all the registered classes, the methods they support, and the parameters they require. CORBA calls these descriptions method signatures. The interface repository is a run-time database that contains a machine-readable version of the IDL-defined interfaces. The APIs let your programs access this information.
- The ORB interface consists of a few APIs to local services that may be of

interest to an application (for example, mapping object references to strings and vice versa).

The client and server components communicate using the ORB core, which is implementation-specific (that is, not defined by CORBA). The idea is that CORBA provides the interfaces above the ORB Core that mask the differences between vendor implementations. Vendors may use any transport they want for the networked ORB traffic and for the basic representation of objects in their system. Sun's DOE uses the ONC RPC over TCP/IP; HP's DOMF uses DCE RPC; and IBM's DSOM supports sockets over TCP/IP, NetBIOS, or IPX/SPX (IBM is also working with HP on a DCE-based ORB). CORBA 2.0 (due late 1994) is working on a solution for multivendor ORB interoperability. The specification, however, should have minimal impact on CORBA 1.1-compliant applications.

CORBA-defined components appear on the server side between the ORB core and the object implementations. Here's a quick explanation of what the pieces do:

- The server IDL stubs (also known as skeletons) provide the static interfaces

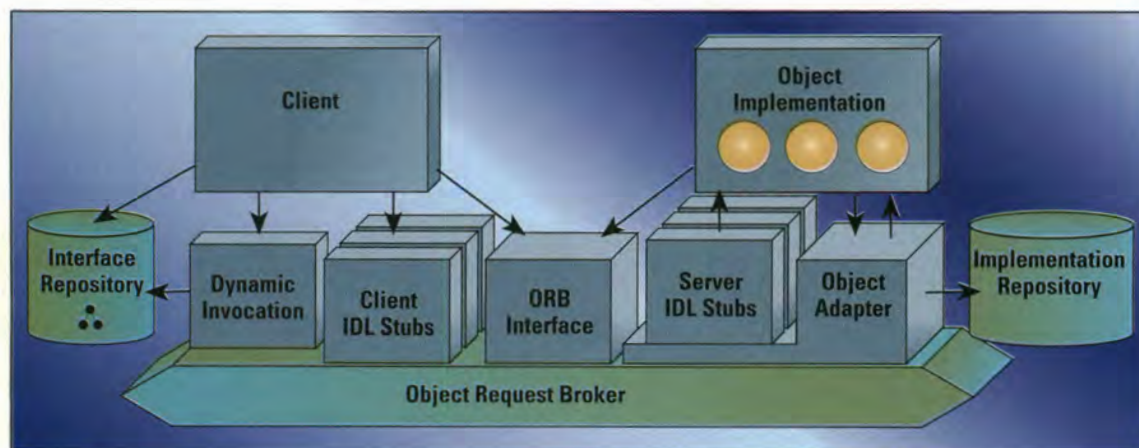


Figure 2. Structure of a CORBA ORB

to each service exported by the server. These stubs, like the ones on the client, are created using the IDL compiler. The server cannot tell the difference between incoming static or dynamic invocations. They're both invoked through server stubs.

- The object adapter interfaces with the ORB's core communication services and accepts requests for service on behalf of the server's objects. It provides the run-time environment for instantiating server objects, passing requests to them, and assigning them object IDs—CORBA calls them object references. The object adapter also registers the classes it supports and their run-time instances (that is, objects) with the implementation repository. CORBA specifies that each ORB must support a standard adapter called the basic object adapter. Servers may support more than one object adapter.
- The implementation repository provides run-time directory information about the classes a server supports, the objects that

are instantiated, and their IDs. It also serves as a common place to store additional information associated with the implementation of ORBs. Examples include trace information, audit trails, security, and other administrative data.

This concludes our panoramic overview of the ORB components and their interfaces. If you are confused, keep reading—the next level of detail may help clarify things.

CORBA CLIENT/SERVER REQUESTS

Figure 3 shows the two types of client/server invocations that are supported by a CORBA ORB: static and dynamic. In both cases, the client performs a request by having access to an object reference (that is, object ID) and invoking the method that performs the service. The dynamic and static interfaces for performing the service satisfy the same request semantics. The receiver of the message cannot tell how the request was invoked. In both cases, the ORB locates a server object adapter, transmits the parameters, and transfers control to the object implementation through the server IDL stub (or skeleton).

Clients see the object interfaces through the perspective of a language mapping—or binding—that brings the ORB right up to the programmer's level. Client programs should be able to work without any source changes (on any ORB that supports the language binding) with any object instance that implements the interface. The implementation of the object, its object adapter, and the ORB used to access it is totally transparent to both static and dynamic clients.

THE CORBA PROCESS: FROM IDL TO INTERFACE STUBS

Figure 4 shows the steps you go through to create your server

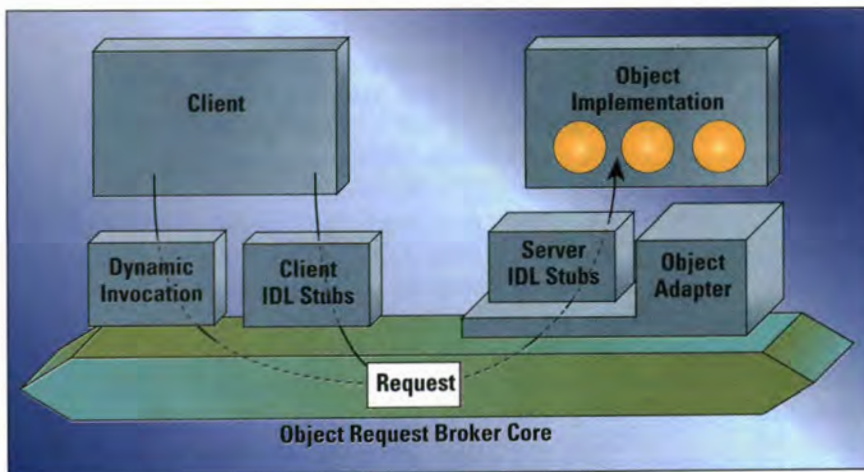


Figure 3. The ORB-based static and dynamic invocation services

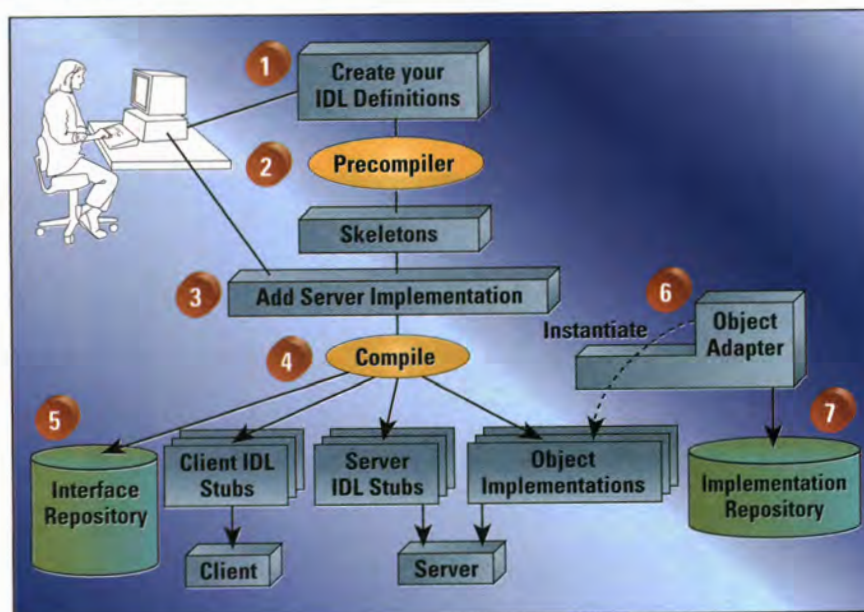


Figure 4. Defining services: from IDL to interface stubs

TRANSITIONING TO SMALLTALK TECHNOLOGY?
INTRODUCING SMALLTALK TO YOUR ORGANIZATION?
Travel with the team that knows the way...

THE OBJECT PEOPLE

Your Smalltalk Experts



SMALLTALK

NEW: VisualAge & IBM Smalltalk
VISIT US
at OOPSLA
Booth #606/608

Education & Training

Smalltalk/V
VisualWorks
VisualAge
ENVY/Developer
Analysis & Design
Project Management
In-House & Open Courses

Project Related Services

Object Immersion Program
Project Mentoring
Custom Software Development
Legacy Systems
GUI's, Databases
Client-Server

The Object People Inc. 509-885 Meadowlands Dr., Ottawa, Ontario, K2C 3N2
Telephone: (613) 225-8812 FAX: (613) 225-5943

Smalltalk/V is a registered trademark of Digitalk, Inc.
VisualWorks is a trademark of ParcPlace Systems Inc.

VisualAge is a registered trademark of IBM.
ENVY is a registered trademark of OTI Inc.



classes, provide interface stubs for them, store their definitions in the interface repository, instantiate the objects at run time, and record their presence with the implementation repository. Let's go through these steps one by one and see what is involved.

1. *Define your object classes using IDL.* IDL is the means by which objects tell their potential clients what operations are available and how they should be invoked. The IDL definition language defines the types of objects, their attributes, the methods they export, and the method parameters. The CORBA IDL is a subset of ANSI C++ with additional constructs to support distribution. IDL is purely a declarative language. It uses C++ syntax for constant, type, and operation definitions; however, it does not include any control structures or variables.

2. *Run the IDL file through a language precompiler.* A typical CORBA-compliant precompiler processes the IDL files and produces C or C++ language skeletons for the implementation server classes.

3. *Add the implementation code to the skeletons.* You must supply the code that implements the methods in the skeletons. In other words, you must create your server classes.

4. *Compile the code.* A CORBA-compliant compiler is typically capable of generating at least four types of output files:

- Import files that describe the objects to an interface repository
- Client stubs for the IDL-defined methods—these stubs are invoked by a client program that needs to statically access IDL-defined services via the ORB
- Server stubs that call the meth-

ods on the server—they're also called up-call interfaces

- The code that implements the server classes.

The automatic generation of stubs frees developers from having to write them and frees applications from dependencies on a particular ORB implementation.

5. *Bind the class definitions to the interface repository.* Typically, a utility is provided to bind—or, if you prefer, compile—the IDL information in a persistent store that can be accessed by programs at run time.

6. *Instantiate the objects on the server.* At startup time, a server object adapter may instantiate server objects that service remote client method invocations. These run-time objects are instances of the server application classes. CORBA specifies different object adapter strategies that are used to create and manage the run-time objects.

7. *Register the run-time object with the implementation repository.* The object adapter records in the implementation repository the object reference and type of any object it instantiates on the server. The implementation repository also knows which object classes are supported on a particular server. The ORB uses this information to locate active objects or to request the activation of objects on a particular server.

The seven steps we just outlined are typical of most CORBA implementations. CORBA, of course, allows deviations. For example, it is not a requirement for IDL source code to be available as long as the interface information is available in stub form or in an interface repository. It is not necessary for the server objects to be implemented as classes as long as they are encapsulated by IDL stubs. The separation of the interface from the implementation

ORB vs. RPC

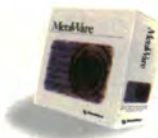
"Brokers of all types—stock brokers as well as object brokers—exact a price for their services."

Bill Andreas,
Chief-Architect, HyperDesk

How are ORB method invocations different from RPCs? The mechanisms are very similar, but there are some important differences. With an RPC, you call a specific function (the data is separate). In contrast, with an ORB, you're calling a method within a specific object. Different object classes may respond to the same method invocation differently through the magic of polymorphism. Because each object manages its own private instance data, the method is implemented on that specific instance data.

ORB method invocations have "scalpel-like" precision—the call gets to a specific object that controls specific data and implements the function in its own class-specific way. In contrast, RPC calls have no specificity—all the functions with the same name get implemented the same way. No differentiated service here. Of course, the ORB is usually built on top of an RPC service, so you end up paying a performance penalty for this "refined" level of service. It is worth every penny if you're taking advantage of new levels of distributed granularity provided by objects. Otherwise, you just bought yourself another layer of middleware—with all the costs and headaches that come with it.

BREAK AWAY FROM THE COMPETITION



When you're pressing the limits of software performance, every microsecond counts. Efficient operation decreases execution time, putting you ahead of the pack.

That's why professional C/C++ developers use High C/C++. MetaWare's High C/C++ compilers offer eight levels of global optimization, plus specific optimizations for particular processors. The executables consistently out-perform competitors in benchmark testing.

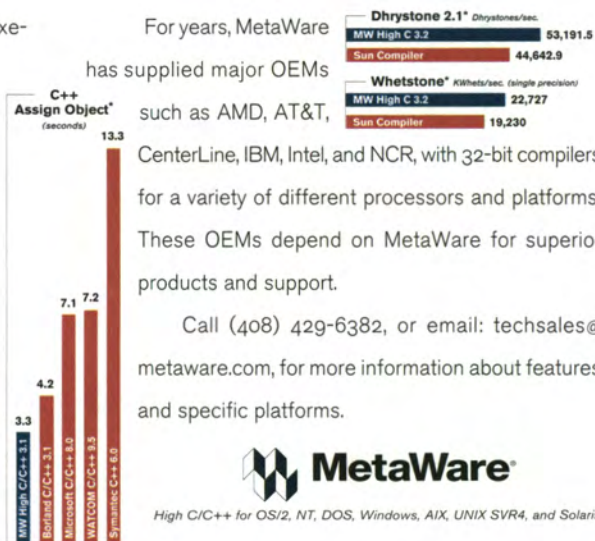
Besides offering rapid execution time, MetaWare's High C/C++ compilers provide a myriad of controls that allow you to customize and fine-tune code generation. There are command-line toggles and pragmas for customizing your application.

The compilers generate informative error and warning messages, and comply with major industry standards.

For years, MetaWare has supplied major OEMs

such as AMD, AT&T, CenterLine, IBM, Intel, and NCR, with 32-bit compilers for a variety of different processors and platforms. These OEMs depend on MetaWare for superior products and support.

Call (408) 429-6382, or email: techsales@metaware.com, for more information about features and specific platforms.



High C/C++ for OS/2, NT, DOS, Windows, AIX, UNIX SVR4, and Solaris

*Benchmark source and other performance information available upon request. All benchmarks use the -O option. MetaWare, the MetaWare logo, High C, and Professional Pascal, are registered trademarks, and High C/C++ and High C++ are trademarks, of MetaWare Incorporated. Other names are trademarks of their respective companies. Copyright 1994 MetaWare Incorporated, 2161 Delaware Avenue, Santa Cruz, CA 95060-5706.

Circle Reader Service Number 39

makes it possible to incorporate existing (legacy) systems within an ORB environment.

HOW METHODS ARE DYNAMICALLY INVOKED

CORBA's dynamic invocation APIs allow a client program to dynamically build and invoke requests on objects. The client specifies the object to be invoked, the method to be performed, and the set of parameters through a call or sequence of calls. The client code typically obtains this information from an interface repository or a similar run-time source. The dynamic invocation provides maximum flexibility by allowing new object types to be added to the distributed system at run time. Visual tools will use this API to create interfaces through point-and-click interactions with a user.

To invoke a dynamic method on an object, the client must perform the following steps, which are shown in Figure 5:

1. *Obtain the method description from the interface repository.* CORBA specifies about 10 calls for locating and describing objects within the repository. After an object is located, a `describe` call is issued to obtain its full IDL definition.

2. *Create the argument list.* CORBA specifies a self-defining data structure for passing parameters, which it calls the `NamedValue` list. The list is created using the `create_list` operation and as many `add_arg` calls as it takes to add each argument to the list.

3. *Create the request.* The request must specify the object reference, the name of the method, and the argument list. The request is created using the CORBA

`create_request` call.

4. *Invoke the request.* The request may be invoked in one of three ways:

- The `invoke` call sends the request and obtains the results.
- The `send` call returns control to the program, which must then issue a `get_response` or `get_next_response` call.
- The `send` call can be defined to be "one way"; in this case, no response is needed.

As you can see, it takes effort to dynamically invoke a method. You're trading off complexity and performance for added flexibility.

STATIC VS. DYNAMIC METHOD INVOCATIONS

The static interface is directly generated in the form of stubs by the IDL precompiler. It is perfect for programs that know at compile time the particulars of the operations they will need to invoke. The static stub interface is bound at compile time and provides the following advantages over the dynamic method invocation:

- It is easier to program. You call the remote method by simply invoking it by name and passing it the parameters. It's a very natural form of programming.
- It provides more robust type checking. The checking is enforced by the compiler at build time.
- It performs well. A single API call is issued to the stub, which takes it from there.
- It is self-documenting. You can tell what's going on by reading the code.

In contrast, the dynamic method invocation provides a more flexible environment. It allows you to add new classes to the system without requiring changes in the client code. It is very useful for tools that discover what services are provided at run time. You can write some very

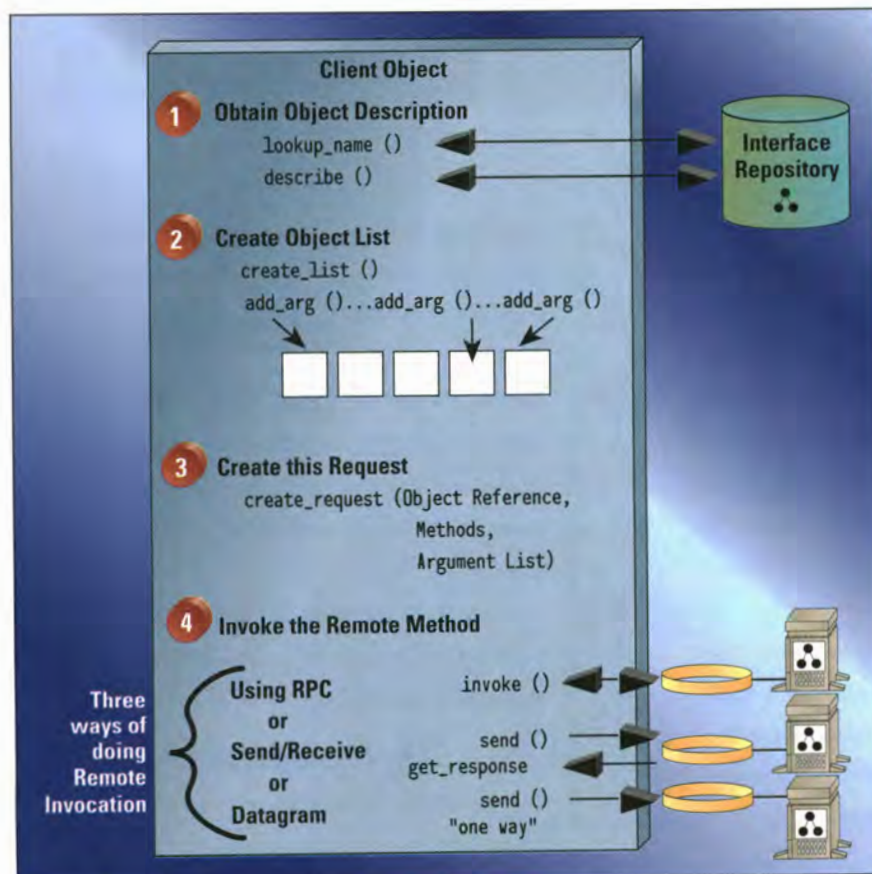


Figure 5. The CORBA dynamic invocation interface

generic code with dynamic APIs. However, most applications do not require this level of flexibility and are better off with static stub implementations.²

Bob Orfali and Dan Harkey are the authors of the best-selling book *Client/Server Programming with OS/2* (VNR, 1993). Bob and Dan's most recent book is the *Client/Server Survival Guide with OS/2* (VNR, 1994). This 930-page book contains over 150 pages on distributed objects. Bob and Dan have been developing client/server applications and tools for the last

eight years. They currently work on the application of distributed object technology. Bob and Dan are affiliated with IBM

Personal Software Products (PSP), Austin, Texas. They work from the San Francisco Bay Area.

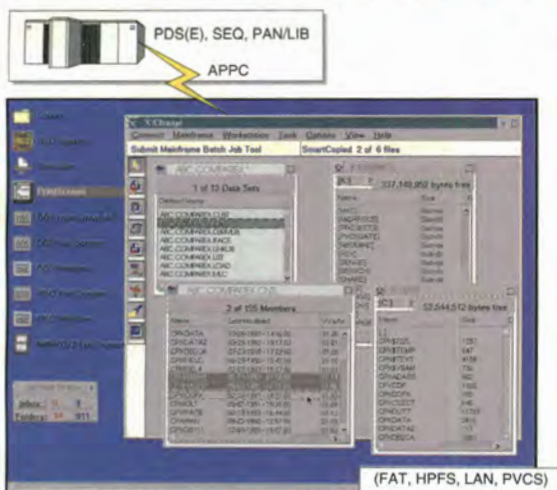


REFERENCES

1. Cliff Reeves, Robert Orfali, and Dan Harkey, "A Distributed Object Road Map for Developers," *OS/2 Developer* (September/October, 1994).
2. Robert Orfali and Dan Harkey, "Client/Server Survival Guide with OS/2" VNR, 1994, IBM publication number SR28-5494. Part 7 provides a 140-page introduction to distributed objects.

Mainframe development on the PC just got much faster and easier

Introducing X:Change, the Cross-Platform File Manager



X:Change brings point and click simplicity to your mainframe-to-PC file management needs.

SERENA is also the developer of COMPAREX®, Change Mail®, PnTxxxx™, and SyncTrac®. X:Change is a trademark of SERENA International. Other product names may be trademarks of their associated companies.

X:Change offers many ways to improve programmer productivity and reduce the cost of offloading mainframe development onto the workstation.

- ✓ No TSO! Bypass TSO SEND/RECEIVE and IND\$FILE completely, for fast mainframe access. X:Change's native MVS/VTAM started task and APPC/LU6.2 communication supports hundreds of users in a single address space, with speeds five times that of TSO for upload/download.
- ✓ Manage, compare, synchronize, and transfer files between multiple MVS and OS/2 locations with the ease of point and click, drag and drop.
- ✓ Simplify access to all major mainframe file types and library managers via a single GUI (PDS[E], Sequential, Parvalet and Librarian).
- ✓ Leverage your investments in other products from other providers such as Micro Focus and INTERSOIX.
- ✓ Utilize your existing security package (RACF, CA-ACF2, CA-Top Secret and SAF) to authorize your user ID and password.

For more information about how X:Change can make a difference in your organization, call us today, toll free:

800-457-3736

Outside the US and Canada call 415-696-1800 - FAX: 415-696-1776



Circle Reader Service Number 40



Object-Oriented Programming

This article shows you how to construct visual wrappers for existing APIs by using VisualAge, IBM's new Smalltalk. **By ROBERT LINDSAY**

Hints for VisualAge Wrappers

Recently, I was part of a team at AMS that used IBM's new object-oriented application development tool, VisualAge. This product supports a visual programming interface and allows IBM Smalltalk to be used for nonvisual components. Our project created a visual parts library by building a set of wrappers for an existing subsystem. The visual parts library allows application developers to use the services of the subsystem through visual programming.

Although many features of VisualAge deserve mention, our intention here is to provide some hints on how to successfully construct visual

wrappers for existing API's. In this manner, you can extend the usefulness of your current subsystems into the visual programming domain.

WRAPPERING

Figure 1 indicates the basic theory behind building wrappers. An interface object is constructed to provide encapsulation and isolation between the object-oriented application and a procedural subsystem. Through isolating low-level services and interfaces, wrappers provide useful building blocks for the object-oriented programmer. Wrappers also limit the exposure of the interface to the internals of a single object and thereby promote reuse. On the downside, wrappers may not provide as balanced a partitioning of function as a completely object-oriented design.

In the VisualAge environment, visual wrappers need to provide a slightly larger service, that of being a useful visual building blocks. This requires a modified approach to usual object-oriented construction techniques because visual building blocks communicate through events connected to actions. Nonvisual object-oriented systems normally communicate via messages between objects. Although this may seem to be splitting hairs, the difference gives

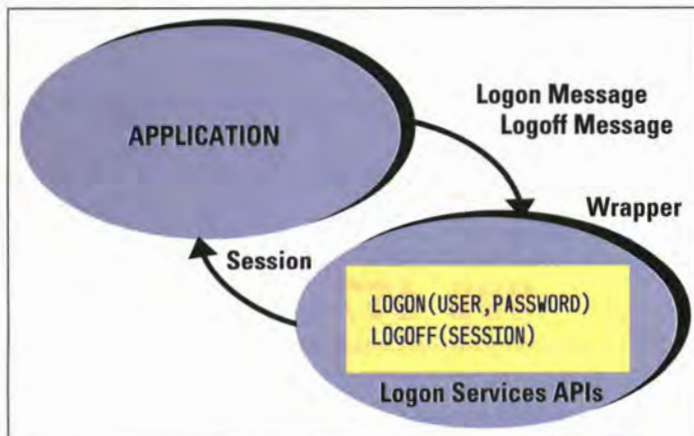


Figure 1. Wrappers build encapsulation



the application programmer an orientation toward either real-time asynchronous processing or controlled synchronous processing.

In this article, we will focus on a simple example: a wrapper that allows a programmer to visually use logon/log-off services invoked via C function calls. This is a very small portion of the functionality available through using wrappers and should not be considered as even typical of the complexity of normal wrappers. It will illustrate our major points, however:

- A modified form of CRC may be used to design parts.
- Visual building blocks should be visually proofed.
- Low-level interfaces can be easily wrapped.

MODIFIED CRC

In our project, we started by first designing what our visual parts library would look like. In so doing, we constructed requirements from various uses of the parts. We then built “paper” parts and performed CRC modeling—a standard object-oriented design technique—to determine if our organization of the parts was correct.

CRC is a technique for validating a proposed clustering of functionality into objects, by reviewing the responsibilities and dependencies of each object via a role playing process. In the course of doing CRC, the balance of behavior assigned to objects becomes apparent, so CRC facilitates a well-balanced design of objects. CRC also provides a documentation method for the eventual ob-

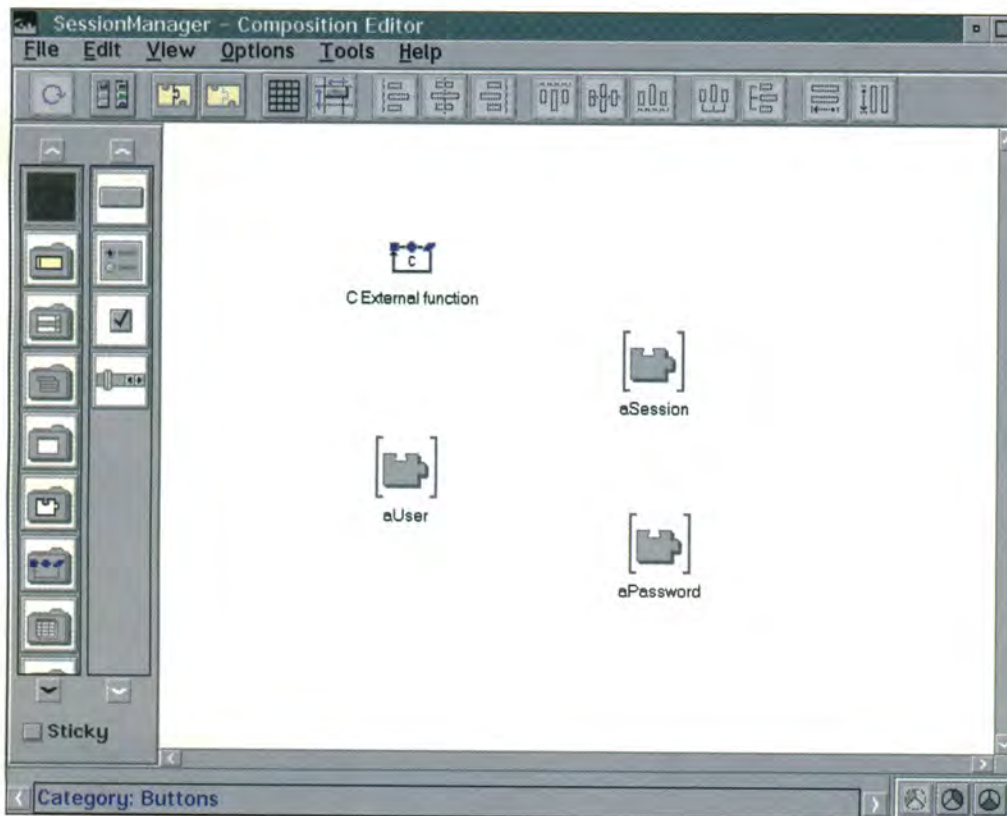


Figure 2. Mock nonvisual part components

ject messages and attributes.

Because of the event-oriented nature of VisualAge programming, we have discovered that a modified form of CRC can work better. In the modified form, we list EVENTS in addition to ACTIONS and ATTRIBUTES. This modification allows for a more explicit paper representation of the functional nature of the part and provides a better indication of the actual role played by the parts in implementing an application.

THE VISUAL PROOFING

Because the parts need to be visually useful, in addition to using a modified CRC, it is also advisable to create mock components and role-play the use of the parts visually to determine if the correct balance between functionality, flexibility, and form have been achieved. This is done by creating mock parts in VisualAge and building several prototype uses of the part.

Constructing mock parts with

VisualAge is straightforward. The parts designer can build these parts in a few minutes and place them onto a visual palette. By providing dummy placeholders of various types, mock parts can be given attributes. Events and actions can also be built into mock parts so they appear to act complete, as shown in Figures 2 and 3.

When mock parts are assembled into a mock application, it becomes easy to identify (if the correct level of parts design has been achieved) even though no real code has been constructed, as shown in Figure 4. Because of the nature of visual programming, some idea of how parameters affect behavior needs to be addressed. For basically static parameters, a notebook control for the part itself is ideal, allowing the programmer to specify values at part selection time. For more dynamic parameters, explicit attributes will need to be defined, and

script language (Smalltalk-like) may need to be written.

At this point, it is a good idea to document and review the steps that were required to use each part to determine if the use is awkward or natural. Ideally, a programmer should find the composition of parts is intuitive, although this may be difficult to get exactly correct when building wrappers. If the parts are built along subsystem functional groupings (for example, I/O operations together in a logical I/O object), it may require some hidden classes to support shared data items, but the programmer will have a reasonable visual metaphor for the underlying subsystem.

When we did this exercise for our subsystem wrappers, we found that fewer, more monolithic parts with several modes of operation were easier to use in building applications than a lot of small parts that had to be assembled

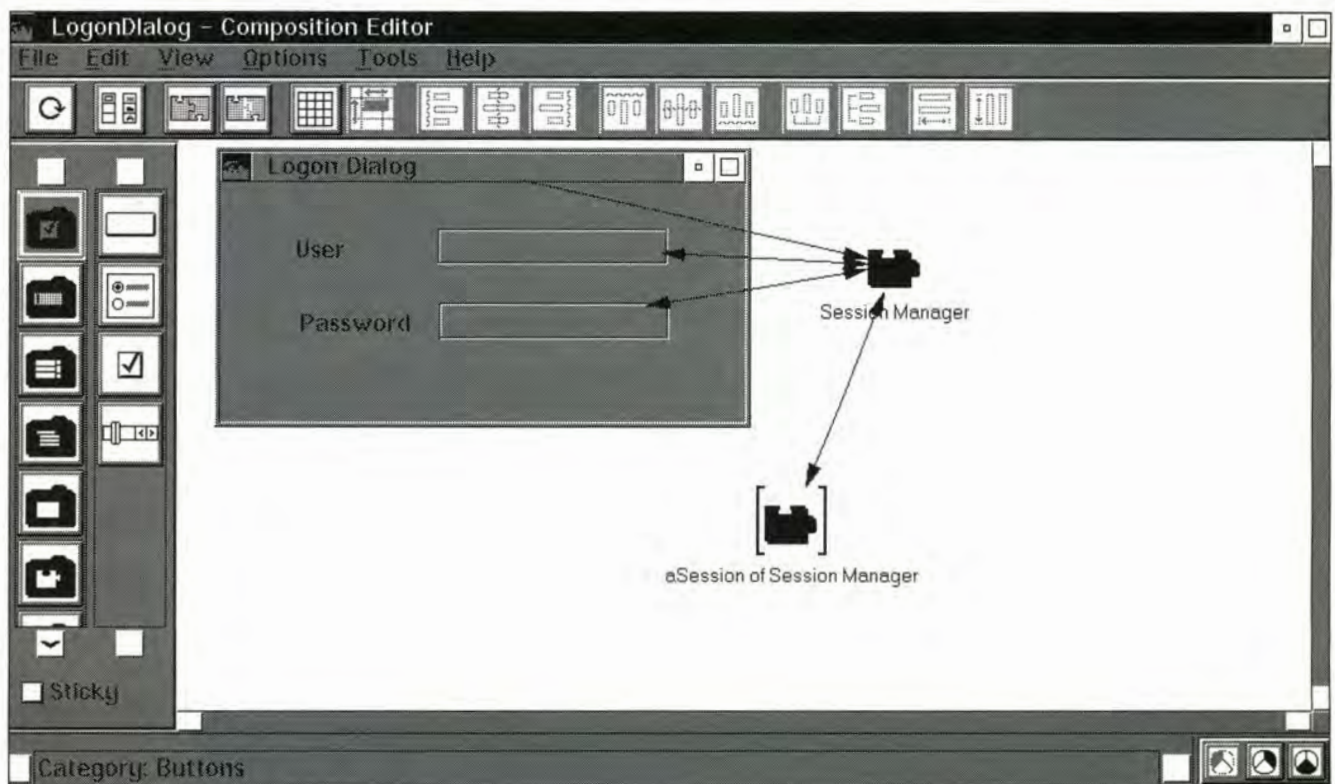


Figure 3. Mock visual part components

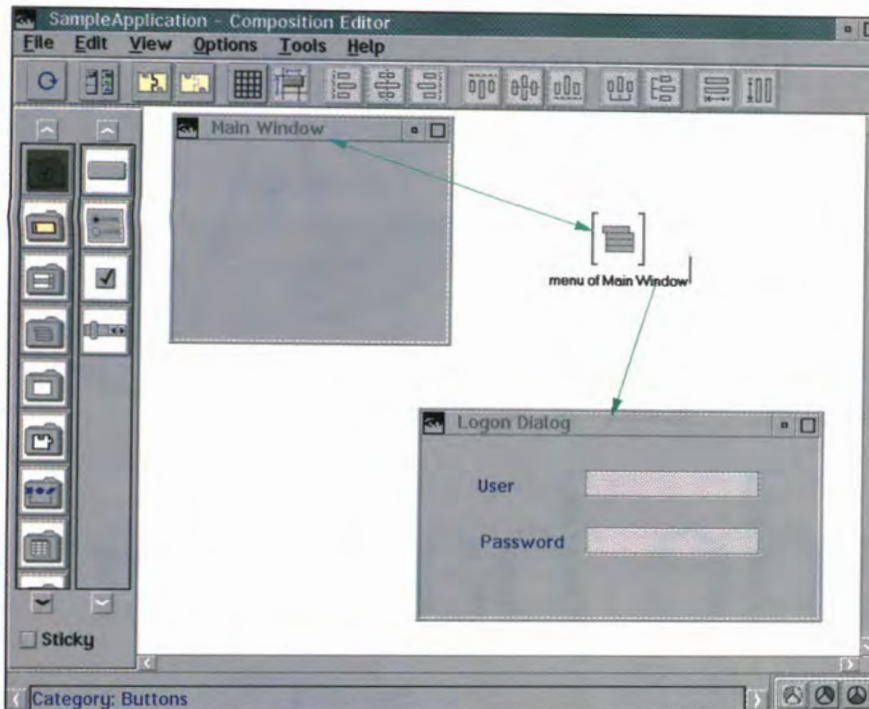


Figure 4. Prototype application using mock parts

```

logoff
    "Perform the Logoff action."
    | pRC |

    pRC := SystemStruct new.

    LindsayLogoff
        coroutineCallWith: aSession with: 0 with: pRC asAddress.

    ^self

```

Figure 5. A coroutine call

```

callLindsayLogon: pszUserID
    with: pszPassword
    with: pRC

    "method to invoke the logon function"
    | |

    <c: int32 'LIBRARY':LindsayLogon pointer
        pointer
        pointer>

    ^self primitiveFailed!

```

Figure 6. A primitive-level direct call

over and over by the application programmer.

After the visual design exercise, the mock parts can be turned over to a parts artist who can then create the desired visual representation of the underlying functionality through good icon design.

LOW-LEVEL INTERFACES

While most of the development effort under VisualAge is straightforward, one low-level situation may require caution. A typical interface from Smalltalk to C is shown in Figure 5. This interface builds a parameter list and then invokes C using built-in IBM Smalltalk functions. Coroutines are used because they allow the visual components to proceed even if the invoked services consume substantial CPU cycles or have blocked threads. This is normally a good default method to invoke low-level services.

In some cases, however, you need to be careful when invoking low-level services using coroutines. These are the calls that may have their own multithreaded structure internal to the API. For some of these API's, control may return to Smalltalk (which will terminate the extra coroutine thread) before the internal threads have completed. The results are then unpredictable.

To avoid this situation, invoke services that create additional internal threads using a simple call interface, as shown in Figure 6 (an illustration of using a Smalltalk primitive to call a C interface).

CONCLUSION

VisualAge's ability to allow the rapid prototyping of systems, followed by the construction of the underlying components, proves to be of great value. We were able to rapidly assess the ease of use of our proposed visual parts with only a minimal construction effort. Once we had verified the design of our

You Discovered the 32-Bit Power of OS/2.

FOR CORPORATE
DEVELOPERS BUILDING
SOLUTIONS ON
OS/2!

Now Master It at OS/2 DEVELOPMENT '95.



Introducing the conference for advanced software development.



he publishers of OS/2
Developer magazine have
assembled industry leaders to host the first
independent Pan-European conference on
OS/2. You'll sharpen your OS/2 skills
and learn from the experts about the latest
techniques and technologies to increase
your effectiveness on the job.

What's included:

- Excellent educational seminars
- Keynote speakers
- Product Roundtables
- And special events!



You'll Learn More About

- SOM programming
- Workplace shell programming
- User interface development
- OpenDOC
- Object-oriented programming
- REXX programming
- SmallTalk, and more!

Produced by

**OS/2
Developer**
magazine

May 8-10, 1995
Krasnapolsky Hotel
Amsterdam

To find out more about this
exciting OS/2 event, fax,
phone, write, or contact us by
electronic mail now!

Telephone:
32-2-538.60.40

CompuServe:
71673,1163

Internet:
CSP@mfi.com

Mail:
OS/2 Development '95
Chaussée de Charleroi 123a
Bte 5
B-1060 Brussels, Belgium

mfi MillerFreeman, Inc.
A MEMBER OF THE UNITED NEWSGROUPS GROUP

OS/2
DEVELOPMENT '95

**SEND BACK THIS
COUPON TO
FAX 32-2-537.56.26**

Name

Title **Company**

Mailing Address:

Telephone

Fax **Email**



visual parts and that they would be easy to use in building the desired systems, we found that fewer more monolithic, parts with several modes of operation were easier to use than a lot of small parts.

This article has presented some of the considerations for constructing VisualAge wrappers for existing services. I would like to thank Gordon Sheppard of AMS for his ideas and contributions in the areas of event-based CRC and other topics. Hopefully, this brief introduction to VisualAge wrappers will assist others in building visual programming environments.

Robert Lindsay, American Management Systems Inc., 4050 Legato Rd., Fairfax, Va. 22033, e-mail: robert_lindsay@mail.amsinc.com. Lindsay is an AMS

senior principal, providing design and advanced development support. He produced many key portions of AMS's CORE foundation software. Prior to joining AMS,

he developed and marketed a COBOL compiler for the IBM Series/1. Lindsay holds a B.S. in electrical engineering from the Massachusetts Institute of Technology.

REFERENCES

Introduction to Object Oriented Programming with IBM SMALLTALK IBM order number SC34-4491.

Guide to the IBM SMALLTALK Development Environment - IBM order number SC34-4492.

IBM SMALLTALK Programming Reference - IBM order number SC34-4493.

VisualAge Users Guide and Reference - IBM order number SC34-4490.

Construction from Parts Architecture: Building Parts for Fun and Profit - IBM order number SC34-4488.

OS/2 Visual Programming with GpfRexx™

Using expertise gained from years of creating the best Visual Programming Tools for C programmers, Gpf Systems has created a visual programming tool for everyone. By combining the easy to use but powerful WYSIWYG Interface Builder of Gpf with the easy to use but powerful REXX language that comes with every OS/2 system, now you can create your own PM applications, even if you've never used Gpf or REXX before. Point, click, drag, and drop to build striking PM applications that take advantage of the advanced features of OS/2.

Visual Programming

Visual GUI programming for REXX; the most natural way to program a Graphic User Interface.

Simple but Powerful

Simple to use but powerful WYSIWYG Interface Builder which can create basic or advanced applications without detailed system level knowledge.

Getting the Most from OS/2

Take advantage of advanced OS/2 features: Multi-Tasking, SQL DataBase, Multi-Media, Multi-Media Controls, Communications (APPC, CPI-C, EHLLAPI)



Order GpfRexx today for Just \$247.50
60 days money back guarantee
For information on other development products call,
Phone: (800) 831-0017 or Fax: (203) 873-3302
30 Falls Road, Moodus, CT 06469



Risk FREE

FEATURES

- Point and click programming
- Multi-thread programming
- Nested menus to any level
- Fully integrated true WYSIWYG interface builder
- PM programming without system knowledge
- Support for CUA 91 controls
- Supports Multi-Media at design and run time
- Supports user designed custom controls
- Control fonts and colors for windows and controls
- Easy inclusion of Bitmaps and graphics
- Context sensitive help for windows and controls
- Point and click access to DB2/2 SQL databases - and REXX supported Embedded SQL access to DB2/2 APIs
- Multi-thread source level debugging
- Royalty FREE - Configurable run-time

The Path To A Good GUI Isn't Always Clear

Navigate the maze with the NEW Gpf 2.1 Professional Developers Toolkit

Promises are cheap, GUI development tools aren't. Gpf 2.1 demo software is FREE. Try Gpf before you buy any tool and you'll agree that Gpf 2.1 DELIVERS.

With this powerful point and click visual programming environment you can create a CUA '91 Graphical User Interface for OS/2 Presentation Manager or MS DOS/Windows® in as little as 10% of the time required to hand code the same design.

What You See Is What You Get programming reduces weeks of coding to hours. Quickly prototype and test your interface, then let Gpf write the code. Gpf generates error free, structured Object Oriented C or C++ source, complete with embedded SQL for OS/2 DataBase. Custom Help and Logic are added to controls as they are drawn to create full Client/Server or Stand-alone applications!

Gpf is hosted on OS/2 2.x and generates native code for OS/2 2.x, OS/2 1.3.x and MS Windows 3.0 or 3.1 Of course this means maximum performance with no run time module and no royalties!

* GUI Programming Facility

Seeing Gpf is believing!

- Design 32 or 16 bit GUIs for IBM OS/2® or Microsoft Windows®
- Be creative, WYSIWYG design environment
- Full CUA '91 Control set, 32 or 16 bit
- Full application and/or DLL generation
- Reusable Objects
- Automatic documentation
- Minimal time to application delivery



Gpf
SYSTEMS, INC.

Try us, FREE Demo Software Available

Order Gpf Pro Toolkit today for just \$1440⁰⁰

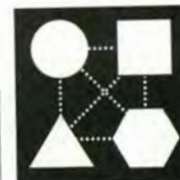
Separately: Gpf 2.1 for \$1295⁰⁰ & GpfTools for \$195⁰⁰

Or, try: Gpf 2.0 Single Platform, now available for just \$495⁰⁰

Call Gpf Systems Inc. at: (800) 831-0017



Phone (203) 873-3300 • fax (203) 873-3302 30 Falls Road, Moodus, CT 06469



*Always wanted to write PM controls but felt it was too much work? Here is a way to alleviate the pain of window proc development using C++ and the IBM Class Library. By **ERIC SNELL** and **LORI RUFFING***

Writing OS/2 PM Controls in C++

The traditional way to write OS/2 Presentation Manager window controls has been to use C and write a window procedure that consists of one large case statement. Any reuse was obtained by a small amount of copy and pasting in an editor, but, for the most part, each new control was basically created from scratch. When programming in a C++ environment, this does not seem to be the correct way to develop new controls: first writing a C control and then wrapping the new control in C++. This article describes a method of writing new Presentation Manager controls in C++ and using inheritance to greatly aid reusability.

GETTING STARTED

When writing a C++ class, our first instinct is to ask: From what class can I inherit to get the basic functionality required? Using the user interface library shipped with C Set++, a likely candidate as parent of a new control would be the ICanvas class. After using ICanvas as the parent for several controls, we came to the conclusion ICanvas was best suited for static controls due to its implementation. The problems with inheriting from ICanvas are as follows:

- Specific class style bits cannot be set since the class is already registered.
 - It is difficult to determine what messages should be allowed to fall through to the canvas. For example, if the canvas sees it is receiving the focus, it will give the focus to another window.
 - Handlers cannot distinguish between subclasses of ICanvas since all subclasses have WC_STATIC as their class name.
 - WM_CONTROL messages will collide. This is related to the previous problem.
- To address these problems, a class specifically designed as a base for new controls was needed.

AN ABSTRACT WINDOW CLASS

The base class we developed for new controls, IAbstractWindow, needed to register a new window class with Presentation Manager to meet some of our design criteria dealing with notifications. A class name, class style, amount of extra window words, and a pointer to a window procedure function are needed for window class registration (see WinRegisterClass). Subclasses of IAbstractWindow supply most of this information during construction, and IAbstractWindow performs the registration.

IAbstractWindow also accepts the stan-

standard ICLUI window creation parameters during construction: window ID, parent window, owner window, initial size and position rectangle, and window style bits. This is so *IAbstractWindow* can create the window and subclass it using the *IWindow::startHandlingEventsFor* method. The constructor registers the new class with Presentation Manager if it isn't already registered through a previous invocation, constructs the window using the *IWindow::create* method, and subclasses the window and adds various default handlers, as shown in Figure 1.

ICLUI presents the concept of handlers, which are classes whose purpose is to receive Presentation Manager messages and invoke virtual functions based on the message content. ICLUI subclasses the de-

fault Presentation Manager window procedure of a control with the *IWindow::startHandlingEventsFor* method. When a Presentation Manager message is received, ICLUI calls a list of handler classes invoking the *IHandler::dispatchHandlerEvent* method. Each handler is added to the window's handler list when it is registered with ICLUI by calling *IHandler::handleEventsFor*. It is the *dispatchHandlerEvent* method that will examine each message and invoke a virtual function in the handler if the message is recognized.

Since the newly registered Presentation Manager class will immediately be subclassed, we can pass *WinDefWindowProc* as the window procedure for all of *IAbstractWindow*'s subclasses. The functionality unique to each window is implemented by overriding the virtual

functions associated with the handler classes. Each message that is not handled will eventually be passed on to *WinDefWindowProc*, just as in the traditional C window procedure.

Default Behavior. Since Presentation Manager windows share many of the same characteristics, *IAbstractWindow* is a good place to implement much of the functionality common to well-behaved controls. For example, each window should send its owner a *WM_CONTROL-POINTER* message when the mouse pointer is over the window. This functionality is implemented in *IAbstractWindow* in the *mouseMove* method and is, therefore, in all subclasses of *IAbstractWindow*. If a subclass needs to do further processing during a mouse move, the subclass overrides *mouseMove*, calls its parent's *mouseMove* method, and performs any processing specific to the subclass.

The set of all messages to which well-behaved controls should respond could be handled in *IAbstractWindow*. Responding to messages such as *WM_MATCHMNEMONIC* and *WM_QUERYDIALOGCODE* can be implemented as virtual functions and overridden in subclasses as needed. This implementation of *IAbstractWindow* may not be complete based on some certain design criteria, but it has enough functionality to allow a subclass to be quickly developed. All that is needed is a constructor and destructor in the subclass.

Many handlers are provided in the ICLUI class library, but not every Presentation Manager message is covered. It was necessary for us to write a handler that would cover some of the functionality we wished to provide. Most handlers group together a set of closely related messages; whereas the handler we wrote, *IAbsWinHandler*, seems to have totally unrelated messages. Since these

```
if (!hasBeenRegistered)
{
    if (WinRegisterClass(IThread::current().anchorBlock(),
                        clsName,
                        WinDefWindowProc,
                        clsStyle,
                        exData))
    {
        hasBeenRegistered = true;
    }
}
if (hasBeenRegistered)
{
    IWindowHandle handle(create(id,
                                0,
                                style,
                                winClassName,
                                parent->handle(), owner->handle(),
                                initial,
                                0,
                                0));

    if (handle.isValid())
    {
        IWindow::startHandlingEventsFor(handle);
        // other handlers added here IAbsWinHandler::handleEventsFor(this);
    }
}
```

Figure 1. Constructor for *IAbstractWindow*



messages are unlikely to be used outside `IAbstractWindow` or its subclasses, we decided one handler would suffice. This example does not illustrate a complete list of messages to which a control should respond. We recommend that a production version of `IAbstractWindow` provides the complete default behavior of a well-behaved Presentation Manager control. This allows for fast, iterative development of Presentation Manager controls.

A NEW CONTROL

Now that we have a base class, developing Presentation Manager controls should be relatively straightforward. The first and easiest step is to write the constructor and destructor of the new control. Being consistent with ICLUI's window constructors, our example control (surprisingly called `NewControl`) takes the familiar window creation parameters. Figure 2 shows how the constructor simply passes the information, along with window class parameters, to `IAbstractWindow`. `IAbstractWindow` does all the underlying work and supplies enough default behavior so we now have a new Presentation Manager control.

Window Styles. `NewControl` is relatively uninteresting with only a constructor, so let's examine the steps taken to start refining the window's behavior. The next step will be to define any style bits the control will have and allow for access and manipulation of these styles. The macro `INESTEDBITFLAGCLASSDEFx` allows you to define a nested `BitFlag` subclass and specify the name of the class associated with and scoping the `BitFlag` class being declared. `NewControl` names the nested class `Style` and also specifies that `IControl` and `IWindow` styles can be combined with `NewControl`'s styles. Another macro, `INESTEDBITFLAGCLASSFUNCS`, de-

fined outside the class declaration, defines the operations that can be performed on the nested `Style` class.

A window class should define a public, static, constant `Style` variable, `classDefaultStyle`, which holds the initial default window styles. It also should define a private, static `Style` variable, `currentDefaultStyle`, which the user can access to change the default style. Class methods `setDefaultStyle` and `defaultStyle` allow the user to set and query the class's default style. The `defaultStyle` method supplies a default value for the `NewControl` constructor. The `setDefaultStyle` method is useful when the user wants to create several controls whose styles are similar but differ from the original class default style.

The individual styles themselves are declared as static, constant `Style` class members in the public section of the class declaration. `NewControl` declares the styles `notifyMB1` and `notifyMB2`. The styles of `NewControl` determine if notifications about mouse button clicks are sent to the owner of the control. These members can be referenced and combined by users of the class to affect the control's behavior. Care should be taken so the chosen style bits do not conflict with predefined Presentation Manager style bits.

Since `classDefaultStyle` and `currentDefaultStyle` are class variables, initial values must be assigned to them. The `classDefaultStyle` is set to some combination of `Style` variables as declared by the classes referenced in `INESTEDBITFLAGCLASSDEF2`. The initial value of `currentDefaultStyle` is set to `classDefaultStyle`. The user can now reference the style constants of the class and treat them like bit flags. An added bonus is that the styles are scoped to the class and their meaning is much more obvious.

Painting the Control. `IAbstractWindow` subclasses need only override the `paintWindow` method to paint the window in response to paint messages. `NewControl` simply paints some text in the center of the window, which displays how many times mouse button one has been clicked in the window. Although `NewControl`'s paint routine is simple, it has much functionality due to its use of presentation parameters.

Window controls should base the colors they use for painting on colors obtained through presentation parameters defined by Presentation Manager. This allows easy control of the colors in a predefined way. In fact, drag-and-drop from the color and font palettes are based on presentation parameters. The presentation

```
NewControl::NewControl(unsigned long id,
                        IWindow* parent,
                        IWindow* owner,
                        const IRectangle& initial, const
                        Style& style)
: IAbstractWindow(newControlClass, newControlClassStyle,
                  0,
                  id,
                  parent,
                  owner,
                  initial,
                  style.asUnsignedLong()),
```

Figure 2. Constructor for `NewControl`

A Perfect Fit Win-OS/2 Support with the Twain for OS/2 Toolkit

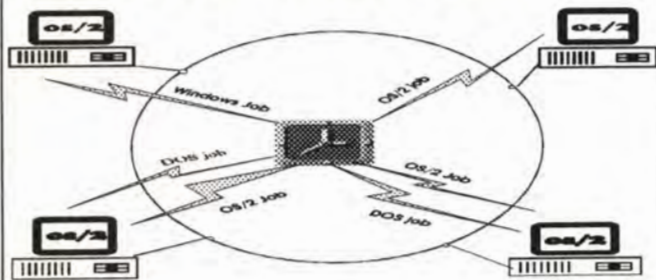


For more information
or
to order your developers kit call:

ST Solution Technology, Inc.
1101 S. Rogers Circle Bldg. 14
Boca Raton, FL 33487
(407) 241-3210 fax (407) 997-6518

Circle Reader Service Number 44

JOB SCHEDULING SERVER FOR OS/2



Presentation Manager 32-bit Job Scheduler for Novell and IBM LANs.

- ✓ Schedulers multiple DOS, Windows, and OS/2 jobs for concurrent execution on the OS/2-based machine.
- ✓ Schedulers jobs on specified or auto selected network nodes.
- ✓ Conditional job scheduling depending on other jobs return codes and presence of multiple files.
- ✓ Complete logging, sophisticated calendar, security, holiday schedule, scheduling same job multiple times, and much more.

Microwork, Inc.

☎ Phone: (708) 940-8979

☎ Fax: (708) 940-8979

Circle Reader Service Number 45

ADVERTISER INDEX

Reader Service	Company Name	Page
22	ACI	38
36	Athena Design	43
	Applications Directory	65
52	Borland International.	COV4
5	Compuware	7
21	Creative Systems Programming	35
	Development '95.	59
20	Development Technologies	33
15	Digitalk	23
9	dSoft Development	14
32	Essex Systems, Inc.	39
2	FTG Data Systems	1
26	GHGRE	38
37	Gpf Systems Inc.	60
6	Great Lakes Software.	9
	IBM	13
	IBM OS/2 Technical Update	29
	IBM Visual Age	57
28	Indelible Blue, Inc..	39
29	IRMware Services	39
23	Katy	38
30	Keller Group	39
39	Metaware.	51
47	MHR Software & Consulting.	75
10	Micro Burst, Inc.	14
3	Micro Focus	2
45	Microwork	64
17	Mitnor Software	26
38	Object People.	49
27	Objectshare Systems.	38
18	On-Line Data.	27
1	One Up Corp.	COV2
31	Op-Tech Data Processing.	39
24	Perez Computing Service.	38
35	Poet Software	41
4	Programmer's Paradise.	4
48	Quadron	79
	Rexx Report	73
51	Rimstar Technology	COV3
	SD '94 East	16
14	SE International	21
40	Serena.	53
49	Softbridge, Inc.	79
13	SofTouch Systems	20
44	Solution Technology	64
33	Solution Technology	37
11	Synaptec	15
12	Token Technology.	17
25	Token Technology.	38
53	Utilis	45
46	Van Nostrand Reinhold.	71
50	Van Nostrand Reinhold.	80
7	Watcom "VXRexx"	11
16	Watcom "Compiler"	25

READY NOW! From the publishers of

OS/2 Developer/Software Development/LAN Magazine/DBMS/
Database Programming & Design/ Dr. Dobb's Journal/UNIX Review and IBM Corp.!

Also Available
On CD-ROM - includes
over 1,000 actual listings
with LAN 3.0 certified products.
Complete electronic brochure for each product.
Made by Selectware™

Now your search for OS/2 solutions just got easier.

Now you can access
the most comprehensive,
detailed listings of vendors
and applications serving
the OS/2 market — all from
one source: the IBM OS/2
APPLICATIONS DIRECTORY.



Applications are conveniently
organized by 17 general industry
categories in alphabetical order:

- Accounting/personal finance
- Business-specific
- CAD/CADAM
- Communications

- Database
- Development tools
- Electronic mail
- Games/entertainment
- Graphics / Desktop Publishing
- Information management
- Integrated workgroup
- Networks
- Spreadsheets
- Utilities
- Word processing

This is the complete guide to solu-
tions for even the most special-
ized applications. It keeps devel-
opers up to date on the competi-
tion and end-users up to speed on
what's available and where to
find it.

The 270-page directory gives you
fully detailed listings that include:

- Company name
- Contact name
- Address
- Phone number
- Product description
- Details on software features
- Price
- Availability date

ONLY \$9.95 FOR BOOK, \$19.95 FOR CD-ROM

☐ Book \$9.95 ☐ CD-ROM \$19.95 ☐ Both \$25.00

Please indicate number of books or CDs desired.

US\$1 shipping/handling for CD-ROM. US\$3.50 shipping/handling for book.

US\$4.50 shipping/handling for CD-ROM and book. Please add US\$2 for shipping to
Canada and overseas.

☐ Payment enclosed (include sales tax in CA, GA, IL, NY & TX; GST in Canada)

☐ Charge my: ☐ Visa ☐ MasterCard ☐ American Express

Card #

Exp. date

Name

Title

Company

Address

City/State/Prov.

Zip/Postal Code

Country

Phone

FAX

Signature (REQUIRED)

Attn: OS/2 Applications Directory

Snyder/Newell Inc. P.O. Box 7046, San Francisco, CA 94120

Call: (800) 444-4881 toll free, or (415) 826-7170

or fax this order to: (415) 826-7378

Plus you get:

- Index of vendors, cross-refer-
enced by product name
- Special separate section of
applications certified by IBM's
Ready! for LAN Server certifi-
cation program

Make your solutions- search easy.

**Send for the IBM OS/2 APPLICATIONS
DIRECTORY.**

OS/2 is a registered trademark of IBM Corp.



space obtained from `IPaintEvent::presSpaceHandle` already has the background and foreground colors established from any previously set presentation parameters; so, basing any other colors on presentation parameters is a natural extension. Painting is not always done in response to paint messages, and some methods default color parameters to specific colors; so explicitly retrieving presentation parameters is good practice (your code is also very explicit with its intentions).

`IWindow` has two methods, `color` and `setColor`, which set specific colors into the window's presentation parameters. The problem with using these methods directly is they take unsigned long parameters to represent the presentation parameters. This is not very meaningful to the user unless he or she does a search of `PMWIN.H`. It is better to define an enumeration that contains all the colors meaningful to

the control. `NewControl` defines an enumeration, `ColorArea`, which contains the item's background, foreground, and hilite. This allows the user to be very sure of exactly what color area he or she intends to set without confusion or mistake, and, of course, the code is extremely readable since we provide a specific type related to our class and not an obscure macro. If you set the value of the enumeration members to the presentation parameters, it makes the code simple and straightforward. Figure 3 shows the `ColorArea` enumeration and the methods dealing with the color presentation parameters.

`NewControl` also calculates the size and position of displayed text each time the control paints. Therefore, the user can drag and drop a font onto the control, and the control will handle the change properly. `IAbstractWindow's` `presParamChanged` method is invoked whenever a presentation param-

eter changes, and its response is to invalidate the window rectangle so the changes will be displayed. A subclass may wish to override this method to do such things as cache color values, precalculate text size and position, or alter its minimum size.

EVENT NOTIFICATION

Most window controls notify their owners of significant events (whether or not an event is significant is defined by the control). The architected method for owner notification in Presentation Manager is the `WM_CONTROL` message. It is a good idea to use the `WM_CONTROL` message for owner notifications since we do not know how our control will be used in the future, and this is the documented method. Also, we would not want to confuse our owner by possibly duplicating another message.

`NewControl` will notify its owner of mouse button one and mouse button two clicks based on the set style bits. The `IWindow::sendEvent` method makes it easy to package this event and send it to the owner. We could leave it at that, forcing the user to understand how the message is encoded, but it is much better to wrap the message in an event and provide a specific handler to catch the event.

`NewControl` Event. Wrapping a specific `WM_CONTROL` event has already partially been done for us. The `IControlEvent` class understands the part of a `WM_CONTROL` message that is not specific to a window class. `IControlEvent` provides a method to determine the ID of the window that sent the message and also a method to return an `IWindow` pointer to the window. It is up to us to subclass `IControlEvent` to provide the methods specific to the `NewControlEvent` class.

While writing event classes for the various controls we are implementing, we found some common

```
enum ColorArea
{
    background    = 3,    // PP_BACKGROUND
    foreground     = 1,    // PP_FOREGROUND
    hilite         = 5,    // PP_HILITE
};

NewControl& NewControl::setColor(ColorArea area, const IColor& color) {
    IWindow::setColor(area, color);
    return *this;
}

IColor NewControl::color(ColorArea area) const
{
    return IWindow::color(area);
}

Boolean NewControl::paintWindow(IPaintEvent& evt)
{
    . . .
    establishColor(clrBackground, background, defClrBackground);
    establishColor(clrForeground, foreground, defClrForeground);
    establishColor(clrHilite, hilite, defClrHilite);
    . . .
}
```

Figure 3. Color enumeration and methods



functions duplicated among the various classes. For a `WM_CONTROL` handler to properly determine which virtual function to invoke, if any, it must know the Presentation Manager class name of the control that sent the message. An application knows the control by its ID, but in the context of a handler, an ID is not meaningful. For example, a `NewControlHandler` must know that an event was received from a `NewControl` window, and the only way this can be determined is by the class name. This called for a new base class for our `WM_CONTROL` message wrappers, `MControlEvent`.

`MControlEvent` provides two more methods, `controlHandle` and `controlClassName`. The `controlClassName` method will be used in each of the handlers we write, including `NewControlHandler`, to match the window class originating the notification to the handler. Now that the window class can be identified, `NewControlEvent` is easily completed. `NewControlEvent` simply provides a method, `mouseNumber`, to extract the information `NewControl` encoded in the message.

NewControl Handler. A handler class provides a set of virtual methods that are invoked based on certain Presentation Manager messages. ICLUI invokes the handler's `dispatchHandlerEvent` method when a Presentation Manager message is received. The `dispatchHandlerEvent` determines if it recognizes the message, possibly constructs a specific event, and invokes a virtual function declared by the handler.

`NewControlHandler` is a handler designed to filter `WM_CONTROL` messages sent from a `NewControl` window. The `dispatchHandlerEvent` method ensures the sender is a

`NewControl` and calls the appropriate `NewControlHandler` method. An application uses this class by adding it to the handler list of the owner of a `NewControl` window. This is the purpose of the `IHandler::handleEventsFor` method. The user overrides the virtual function related to the message to be examined. When that message is received, the user's method is invoked.

CONCLUSION

The `NewControl` example presented here is a simple control but presents many techniques that can be used to simplify and speed development of Presentation Manager window controls. A few well-designed base classes will encourage development of new Presentation Manager classes where developers may have been hesitant. The default functionality of base classes gives subclasses a large head start.

Your new classes are not restricted to use by C++ programmers. With careful design, you can make C++ Presentation Manager

classes available to C and other language programmers with no extra overhead. This is left as an exercise for the reader due to space considerations. Full compilable source code is available on CompuServe's OS2DF2 Forum, OS/2 Developer section. Download file `CONTRL.ZIP`.

Eric Snell is an advisory programmer in OS/2 Multimedia development. His responsibilities include designing and implementing multimedia user interface and subsystem objects. Snell is currently the lead designer/developer for the OS/2 Multimedia WPS/OpenDoc team. He can be reached at 1000 NW 51st Street, Int. Zip 1510, Boca Raton, Fla. 33431, or via e-mail at esnell@vnet.ibm.com.

Lori Ruffing is an associate programmer in OS/2 Multimedia development. Her responsibilities include designing and developing multimedia applications and user interface objects. Ruffing can be reached at 1000 NW 51st Street, Int. Zip 1510, Boca Raton, Fla. 33431, or via e-mail at lori_ruffing@gator.bocaraton.ibm.com.

REFERENCES

"Demystifying Custom Controls," OS/2 Developer Vol. 5 No. 1.

C/C++ Tools, User Interface Class Library User's Guide (S829-3743).

C/C++ Tools, User Interface Class Library Reference (S819-1179).

OS/2 2.0 Programming Guide Vol. II, (S10G-6494).

OS/2 2.0 Presentation Manager Programming Ref. Vol. I, (S10G-6264).

OS/2 2.0 Presentation Manager Programming Ref. Vol. II, (S10G-6265).

OS/2 2.0 Presentation Manager Programming Ref. Vol. III, (S10G-6272).



Buyer's Guide

The staff of OS/2 Developer put together this special section to guide you through the various object-oriented development tools available for OS/2. The products described in this guide are based on surveys sent to vendors; they are provided as a service to you and are free to vendors. These listings do not represent an endorsement by OS/2 Developer. Although every effort has been made to ensure accuracy, we do not assume any responsibility for error or omission in this section.

Object-Oriented Programming Buyer's Guide

SOFTWARE GMBH

Circle No. 101

VisualDesigner is an interactive, graphical software development tool that supports analysis, design, and programming. It creates ready-to-use applications for VisualWorks and generates executable Smalltalk code automatically for the object model, the user interface, and database access. It ensures validation of defined requirements and improves communication between users and programmers. Price: \$4,900.

NEDT is a tool for designing complex graphical editors and is integrated into the VisualWorks environment. The document to be edited can be arbitrary net structures represented by Smalltalk objects. NEDT provides predefined classes that implement behavior for documents, document pages, and network components. There are two add-ons to NEDT. Black-Boxes is a generic graphical editor; potential applications include petri net or chemical processes and any kind of electronic circuit design. Tree is a tool for designing graphical editors. It facilitates the development of editors for tree structures by providing a generic tree editor that can be plugged to applications by simply specifying a few interface messages. Price: \$1,800.

VICK is a tool for interactively building graphical interfaces for VisualWorks. It provides a wide variety of predefined graphical user interface elements for controlling items like bar graphs, function editors, sliders, and turning knobs. Browsers can be used to supervise the state of ob-

jects permanently, and each object can have a multitude of task-dependent user interfaces concurrently. Price: \$980.

ARS NOVA Software GmbH, Stettener StraBe 32/3, 73732 Esslingen, Germany. +49-711-370-4001, fax +49-711-370-4003.

ATHENA DESIGN INC.

Circle No. 102

Mesa 2 is designed for users and developers who want to combine the power and reliability of OS/2 with the flexibility and ease of object technology. You can leverage Mesa's power and flexibility by integrating Mesa objects into your own applications, using the Mesa Object Library Interface (MOLI). MOLI gives developers access to the same spreadsheet objects and features that Mesa 2 itself uses. Mesa 2 has the functionality and ease of use you expect from an advanced 32-bit spreadsheet product. Price: special introductory offer, \$99.

Athena Design Inc., 17 Saint Mary's Court, Boston, Mass. 02146-4007, (800) 315-6372 or (617) 734-6372, fax (617) 734-1130.

BLUE SKY SOFTWARE CORP.

Circle No. 103

WinMaker Pro 6.0, the new version of WindowsMAKER Professional, Prototyper & C/C++ Code Generator for Windows, Win32, and Windows NT doubles C/C++ compiler productivity. WinMaker Pro 6.0 is as visual and easy to use as Visual Basic but provides the developer with industrial strength C and C++ code instead of Basic. Price: \$995.



Blue Sky Software Corp., 7486 La Jolla Blvd., Ste. 3, La Jolla, Calif. 92037, (800) 677-4946 or (619) 459-6365, fax (619) 459-6366.

BORLAND INTERNATIONAL **Circle No. 104**

Borland C++ 1.5 for OS/2 includes support for the latest C++ standards, including exception handling and ANSI string classes, BPMCC control classes that bring the functionality of Borland-style custom controls to OS/2, and ObjectBrowser technology, which speeds up development cycles with graphical representations of the relationships between objects in an application. Price: \$495.

Borland, 100 Borland Way, Scotts Valley, Calif., 95066-3249, (408) 431-1000, fax (408) 431-4175.

CADRE TECHNOLOGIES INC. **Circle No. 105**

ObjectTeam for Shlaer-Mellor 6.0 automates the Shlaer-Mellor methodology for object-oriented development and the production of documentation. Its analysis tool performs automated checking and lets users construct a graphic representation of a system. Its design tool lets designers create and modify design elements graphically. Price: \$4,000+.

Cadre Technologies Inc., 222 Richmond St., Providence, R.I. 02903, (800) 743-2273 or (401) 351-5950, fax (401) 455-6800.

CLASSIC SOFTWARE **Circle No. 106**

Btrv++ 2.51 is a C++ library of 15 major classes providing a C/C++ database interface to Btrieve. It implements all Btrieve operations as member functions and many value-added functions for SQL queries, file and index creation, iteration, extended operations, and global Btrieve management. Complete record and field-based access. It includes an object-oriented C interface, C/C++ interface to Btrieve DDF Data Dictionaries, and 32-bit support. It is a multcompiler platform product supporting Windows, Win32s, Win32, DOS, and

OS/2 (16 and 32 bit). Source included. Royalty free. Price: \$249.

Btrvgen 2.0c is a database schema designer and C++ code generator for Btrieve. It supports all Btrieve data types and has an integrated Btrieve DDF file editor. Generated code supports Windows, DOS, and OS/2. A special bundle is available with Btrv++ and VBtrv\C++ products. Price: \$229.

Classic Software Inc., 3542 Pheasant Run Circle, Ste. 8, Ann Arbor, Mich. 48108, (800) 677-2952 or (313) 677-0732, fax (313) 971-3287.

COMPILER RESOURCES INC. **Circle No. 107**
Yacc++ and the Language Objects Library 2.0

is an object-oriented rewrite of lex and yacc which automatically generates C++ classes of lexer, parser, and abstract syntax tree objects from grammars. The library provides classes for various models of AST, error, input, lexer, parser, and symbol table semantics. Price: \$995.

Compiler Resources Inc., 85 Main St. Ste. 310, Hopkinton, Mass. 01748, (508) 435-5016, fax (508) 435-4847.

DIGITALK **Circle No. 108**

Smalltalk/V for OS/2 is a 32-bit, object-oriented development environment for OS/2. It contains a set of tools for developing graphical, portable CUA-compliant applications and includes a reusable class library, a push button debugger, an incremental compiler, browsers, and inspectors. Price: \$995.

PARTS Workbench for OS/2 is a client/server integration tool that allows application development through easy assembly and reuse of application components. It comes with over 65 prebuilt components, both visual and nonvisual. Other components can be written in Smalltalk/V, C, COBOL, or other languages. Price: \$1,995.

Digitalk PARTS 1.0 (Parts Assembly and Reuse Tool Set) is a component-based application development environment con-



sisting of PARTS Workbench, Smalltalk/V, and Team/V, a team programming and configuration tool. Price: \$5,000.

Digitalk, 5 Hutton Centre Dr., Santa Ana, Calif. 92707, (800) 922-8255 or (714) 573-3000, fax (714) 513-3100.

EASEL CORP.

Circle No. 109

Synchronicity 1.0 is a development tool set for business object management that lets corporate Information Services organizations move to object technology. It integrates the design, assembly, and reuse of business objects to reduce the time required to develop, enhance, and extend strategic enterprise client/server applications. It is integrated with ENFIN.

ENFIN 4.0 is an object-oriented client/server application development environment based on Smalltalk. It provides developers with a spectrum of tools to develop modular reusable applications that can be deployed across a multiplatform network. It has a layer of visual tools to improve programmer productivity and is integrated with Synchronicity. Price: \$5,995 SQL Edition, \$10,900 Corporate Edition.

TeamBuilder 1.0 is a development tool that enables groups of developers to simultaneously build object-oriented applications. Developers can group objects together as a project, with each having a separate revision level, date stamp, and label. It is integrated with Intersolv's PVCS product, providing functionality—archiving older classes and files, comparisons between revisions, and labeling. Price: \$495.

Easel Corp., 25 Corporate Dr., Burlington, Mass. 01803, (800) 625-3281 or (617) 221-2100, fax (617) 221-2196.

FOUR SEASONS SOFTWARE

Circle No. 110

SuperNOVA 3.3 is an object-oriented, platform- and database-independent 4GL that allows applications to run on networks consisting of different operating systems and proprietary databases. Its database independence allows applications to be developed without using database-specific code,

and its platform independence allows graphical user interface programs to run in MOTIF, Open Look, or MS-Windows. It is portable through UNIX, MS-DOS, and VAX VMS. Price: \$1,740 to \$190,000 per CPU depending on configuration.

Four Seasons Software, 2025 Lincoln Highway, Edison, N.J. 08817, (908) 248-6667, fax (908) 248-6675.

FREE UNIVERSITY OF BERLIN

Circle No. 111

Distributed Application Developer's Toolkit 1.0 consists of RPC class library (C++), XDR library, and OS/2 RPC stub generator. It enables inexperienced programmers to implement distributed applications. By applying the stub generator, nondistributed applications can be turned into distributed ones with almost no changes to the source code. The OS/2 Distributed Application Developer's Toolkit is delivered with documented APIs for C++ and numerous examples. Price: \$49 plus shipping.

Free University of Berlin, WRZ, Garystrasse 21, D-14195 Berlin, Germany, +49 308382775, fax +49 308382843.

IBM

Circle No. 112

VisualAge allows the developer to create industrial-strength client/server applications quickly. Both stand-alone and team versions are backed by IBM service and support. It allows you to visually construct applications by selecting and connecting reusable software components, iterate on design, reuse legacy or 3GL, and access multiple databases simultaneously. According to the company, it includes an easy-to-use composition editor.

IBM, 11000 Regency Parkway, Cary, N.C. 27511, (800) 426-2279 or (919) 469-7763, fax (919) 469-7423.

INFERENCE CORP.

Circle No. 113

ART *Enterprise 1.0 is a graphical client/server development tool that uses objects to integrate databases, documents, and business policies. Users access objects with free-form descriptions of the nature or location of the underlying information.

Applications apply policies to volunteer intelligent advice. Example applications include sales force automation, direct marketing, customer service, and scheduling/logistics. Price: \$7,995 including one week of training.

Inference Corp., 500 N. Continental Blvd., El Segundo, Calif. 90245, (800) or (310) 322-0200, fax (310) 322-3242.

INSOFT OY

Circle No. 114

Prosa/pm Object Oriented Analysis and Design Tool 4.0 supports Rumbaugh OMT method. Correctness of models is checked interactively. The user interface of Prosa/om is fast and easy to use, and it supports team development in networked environments. It integrates with Prosac++ code generation, with Prosaql for SQL code generation and with Prosacdm for automated, concurrent documentation. Price: \$3,230.

Prosa/sa Structured Analysis and Design Tool 4.0 supports Yourdon SA/SD/RT method as a system and software specification and design methodology. The models are designed with the same modeling editor, providing an integration of models and interactive checking of models. It integrates with Prosac++ code generator for C code, Prosaql for SQL code, and Prosacdm for automated, concurrent documentation. Price: \$3,230.

Prosac++ C++ Code Generator 4.0 transforms OOA/OOD models designed with Prosa/om into C++ source code and SA/SD/RT models designed with Prosa/sa into ANSI standard C source code. The generated C/C++ code is ready to compile, link, and execute. Its code is portable and compact. Prosac++ supports automatic generation of applications ranging from client/server and database systems to interactive, windows, and real-time systems. Price: \$4,120.

Prosac SQL Code Generator transforms object models designed in Prosa/om OOA/OOD and entity relationship models designed in Prosa/sa SA/SD/RT CASE environments into SQL code. Prosaql checks the models for inconsistencies in

Develop Your Mind—The Best In OS/2 Books!



Object Oriented Programming Using SOM and OSOM

by Christina Lau

The first comprehensive guide to SOM technology and its accompanying frameworks. Especially valuable now that SOM for Windows is available. Answers questions, provides codes and complete implementation of CORBA specifications. Includes disk. \$39.95 0-442-01948-3 IBM #SR28-5570

NEW!



Client/Server Survival Guide with OS/2®

by Robert Orfali and Dan Harkey

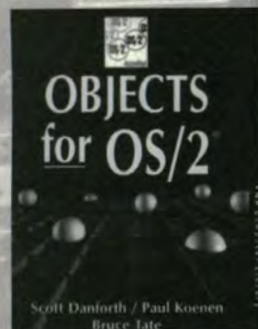
Provides a sweeping tour of client/server and distributed objects. An easy-to-follow guide that comprehensively reviews the technology, standards and over fifty commercial OS/2 client/server products. Offers 969 pages of essential information plus 400 illustrations. \$39.95 0-442-01798-7 IBM #SR28-5494

Client/Server Programming with OS/2® 2.1, 3/E

by Robert Orfali and Dan Harkey

\$39.95 0-442-01833-9 IBM #G325-0650-02

ALSO!

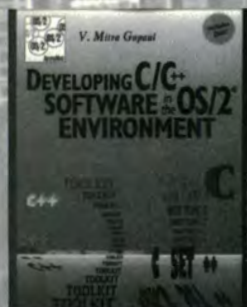


Objects for OS/2® 2.X

by Bruce Tate, Scott Danforth, and Paul Koenen

Integrates the System Object Model with OS/2 2.1 client/server functions. Explains how OOP and SOM function, why they are important, and how they relate to OS/2 2.1. \$36.95 0-442-01738-3 IBM #SR28-5568

NEW!



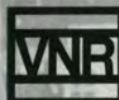
Developing C/C++ Software in the OS/2® Environment

by V. Mitra Gopaul

This important OS/2 programmer's reference combines OS/2 development tools that are scattered through many different IBM manuals into one easy-to-use source. Covers all levels of C and C++ application development in the OS/2 environment using the IBM C Set++ Compiler. \$39.95 0-442-01240-3 IBM #SR28-5569

NEW!

*Ask for these and other VNR titles
at your local bookstore.*



Van Nostrand Reinhold

115 Fifth Avenue, New York, NY 10003

1-800-544-0550

73373.64 @Compuserve.com

OS/2® is a registered trademark of the IBM Corporation.

1645



database definitions, primary and foreign keys, attributes, relationship cardinalities, normalization, and syntax rules of the target DBMS. It is compatible with ISO/IEC SQL standard, DB2, Informix, Ingres, and Oracle database management systems. Price: \$1,175.

Insoft Oy, Prosa Software, Kirkkokatu 5 B, FIN-90100, Oulu, Finland, +358-81-376128, fax +358-81-371754.

INTELLIGENT ENVIRONMENTS

Circle No. 115

Applications Manager 4.3 is a comprehensive, high-level OS/2 development environment for creation of large-scale client/server applications. It helps teams develop graphical user interfaces, logic coding and testing, data integration, communications, quality control, and maintenance. It is designed for scalable, multimodule applications and simplifies mission-critical programming requirements for SQL, transaction management, and legacy system renovations. Price: contact vendor.

Intelligent Environments, 2 Highwood Dr., Tewksbury, Mass. 0187, (800) 669-2797 or (508) 640-1080, fax (508) 640-1090.

INTERACTIVE SOFTWARE ENGINEERING INC.

Circle No. 116

ISE Eiffel for OS/2 3.3 is an object-oriented software development environment for building large, complex, in-house, and commercial applications. It consists of an integrated development workbench, an interactive application and graphical user interface builder, analysis and design CASE tools, database interfaces, and a variety of basic, graphical, and application-specific libraries. Price: \$695-\$995.

Interactive Software Engineering Inc., 270 Storke Rd., Suite 7, Goleta, Calif. 93117, (805) 685-1006, fax (805) 685-6869.

JBA INTERNATIONAL

Circle No. 117

Guidelines 2.1A is an object-oriented OS/2-hosted graphical user interface development environment that automates most of the creation of a graphical user interface applications,

eliminating some traditional steps. Developers are able to create applications ranging from fully event driven procedural to event driven object oriented. It creates native OS/2 and Windows 3.1 applications. Supported servers include the DB2 family, ODBC, SQL-based RDBMS, and over 40 databases through the Q+E libraries. Price: contact vendor.

JBA International, 3701 Algonquin Rd., Crossroads Ctr. Ste. 1000, Rolling Meadows, Ill. 60008, (800) 522-4685 or (708) 590-0299 or in Canada (905) 940-2442, fax (708) 590-0394.

KASE SYSTEMS

Circle No. 118

KASE:C++ is a visual design and code-generation tool for the OS/2 platform. Developers use the designers to create OS/2 objects and associate them with business logic and data. KASE:C++ will then generate the C++ source code. It fully supports IBM's User Interface Class Library. Price: \$1,495.

KASE Systems, 1 Meca Way, Ste. 150, Norcross, Ga. 30093, (404) 564-5696, fax (404) 564-5679.

LIANT SOFTWARE CORP.

Circle No. 119

C++/Views 3.0.1 is an object-oriented application framework for developing multiplatform, native graphical user interface programs using C++. It includes a library of over 100 C++ classes for developing your graphical user interface application. It includes C++/Views Constructor, a visual development tool that unites an interface builder with a class browser. Price: \$999.

Liant Software Corp., 959 Concord St., Framingham, Mass. 01701, (800) 237-1873 or (508) 872-8700, fax (508) 820-0035.

MICADO SOFTWARECONSULT GMBH

Circle No. 120

micWOP Developer 2.0 is a development environment for Smalltalk/V (OS/2 2.x 32 bit or Win32) that has and uses various commercially available Smalltalk/V add-on products. Included are a WindowBuilder, project organizing utilities (ProjektBrowser or Team/V or

Envy), and add-ons developed by micado, such as a translator, a documentation manager, and an environment manager. Price: \$2,710.

micWOP Corporate is a development environment for Smalltalk/V (OS/2 2.x 32 bit or Win32) that extends the micWOP Developer with database classes. It uses the Q+E as well as the ODBC driver, which provides access to most of the commercially available databases: MS SQL Server, Sybase SQL Server, Oracle version 6+7, IBM DB 2/2, IBM DB2-IBM SQL/400-IBMSQL/DS via Gtwy, Gupta, dBase, II+IV, Novell SQL+BTrieve, Excel, and so on. Price: \$5,410.

micWOP (micado Workplace for Smalltalk/V) Corporate ODBMS 2.0 is a development environment for Smalltalk/V (OS/2 2.x 32 bit and/or Win32) that extends the micWOP Corporate with an object-oriented database component. Price: \$8,120.

micDESIGNER 1.0 is a graphical development environment for Smalltalk/V (OS/2 2.x 32 bit and/or Win 32) that enables one to put together a class and method framework. The result is not mnemonic class and method description but a class hierarchy in pure Smalltalk code. Price: contact vendor.

micado SoftwareConsult GmbH, Reutherstr. 1 a-c, 53773 HENNEF, Germany, +49-(0)2242-871-450, fax +49-(0)2242/871-455.

MICRO DATA BASE SYSTEMS INC.

Circle No. 121

Object/1 3.0 is a graphical development environment for DOS Windows and OS/2 Presentation Manager. It offers an object-oriented programming language, forms painter, and open architecture. Access to DMBs such as MDBS IV, SQL Server, Oracle RDBMS, and IBM Database Manager is available. It has over 250 predefined classes, 6,000 predefined methods with source code, full DDE support, and C++-like syntax. Price: \$4,000.

Micro Data Base Systems Inc., 1305 Cumberland Ave., P.O. Box 2438, West Lafayette, Ind. 47906, (800) 445-6327 or (317) 463-7200, fax (317) 463-1234.

A Special Report from the publishers of OS/2 Developer!



Quantities are limited. Order now!

For fastest service, fax your order to us at 415-905-4967



☒ **YES!** Please send me _____ copies of the REXX Report!

Each copy is only \$9.95 in the US or \$10.95 in Canada.

Please add \$2.50 for shipping and handling.

☐ My check is enclosed

☐ Charge my credit card: ☐ Visa ☐ MasterCard ☐ American Express

Name _____

Signature _____

Date _____

Mail: REXX Report
P. O. Box 7046
San Francisco
CA 94120

Phone: 1-800-444-4881

Fax: (415) 905-4967

Send them to:

Name _____

Address _____

City _____

State _____

Postal Code _____



NETWORK TECHNOLOGY CORP.

Circle No. 122

MEDIAscript OS/2 Desktop Edition provides a visual programming environment for the development of object-oriented multimedia applications. It fully supports digital, audio, midi, animation, software and hardware video, ActionMedia II, and MPEG in the OS/2 environment. It is a multi-threaded multimedia tool and provides an icon-based authoring interface for nonprogrammers. Price: \$895.

MEDIAscript OS/2 Professional Edition provides a visual programming environment for the development of object-oriented multimedia applications. The Professional Edition provides an icon-based authoring interface for nonprogrammers as well as the rich MEDIAscript scripting language for more demanding requirements. Price: \$3,995.

Network Technology Corp., P.O. Box 240, Dobbs Ferry, N.Y. 10522-0240, (914) 478-4500, fax (914) 478-4507.

NEURON DATA INC.

Circle No. 123

NEXPERT OBJECT 3.0 is an object- and rule-oriented application development tool. It incorporates business rules and object components that enable you to add simple or complex business rules to make applications "smarter." Using an intuitive visual editor, you enter If-Then-Else rules in any sequence and as separate entities without predefining program flow control. Thus, you can update rules to adapt to changing business conditions. Price: \$8,250.

SMART ELEMENTS 2.0 provides a development environment for generating portable, object-oriented client/server applications that incorporate business rules and objects. A graphical user interface builder component includes a comprehensive set of interface widgets—tables, tree browsers, business graphics, color icons and images, and multifont text. With optional transparent data access component, applications can connect to various data sources. Price: \$13,850.

Neuron Data Inc., 156 University Ave., Palo Alto, Calif. 94301, (800) 876-4900 or (415) 321-4488, fax (415) 321-3728.

OBJECT/FX

Circle No. 124

Visual Companion 1.0 is an object-oriented geographic visualization system that stores and presents information in the form of maps, tables, and images. The system enhances other information products with geographic visualization and analysis capabilities, regardless of source language. Price: \$1,000.

Spatial Works 1.0 is designed for use by object-oriented software developers. This product provides component and framework classes for reuse by those who assemble new object-oriented products that require geographic data management, visualization, and analysis. Price: \$7,500 Development, \$1,000 Run-time.

Object/FX, 2515 Wabash Ave., St. Paul, Minn. 55114, (612) 644-6064, fax (612) 644-6064 ext. 17.

OBJECT INTERNATIONAL INC.

Circle No. 125

Together/C++ allows up-to-date object modeling and C++ programming. Changes in the user's object model appear in code, and added code gets reflected in the object model. It includes view management, filtering, automatic layout, C++ parser, scripted documentation generation, and version control. Price: \$4,400. Special introductory pricing is available for a limited time.

Object International Inc., 8140 North MoPac, 4-200, Austin, Tex. 78759, (800) 662-2667 or (512) 795-0202, fax (512) 795-0332.

OBJECTS INC.

Circle No. 126

Layout 3.0 is an object-oriented application development system that enables users to build applications without using a programming language. Instead, users work with actual on-screen objects, arranging them in a simple diagram to represent their program. Price: \$299.95.

Objects Inc. 99 Rosewood Dr., Danvers, Mass. 01923, (800) 424-

6644 or (508) 777-2800, fax (508) 777-0180.

OBJECTSHARE SYSTEMS INC.

Circle No. 127

WidgetKit/CUA'91 provides the suite of CUA'91 controls for Digitalk Smalltalk/V on Windows and Win32 as well as OS/2 when used in conjunction with WindowBuilder Pro/V. The WidgetKit/CUA'91 controls integrate transparently into the Smalltalk/V and WindowBuilder Pro/V environment. Available controls include NoteBook, Container, Slider, ValueSet, and SpinButton. Source code included; no run-time fees. Price: \$295.

Objectshare Systems Inc., 5 Town & Country Village, Ste. 735, San Jose, Calif. 95054, (408) 970-7280, fax (408) 970-7282.

OBJECTS ++ SOFTWARE CORP.

Circle No. 128

SQL Objects++ Database Library 2.0 is a complete multiplatform object-oriented class library that provides complete database access to many of the major databases with a single API/Method. SQL Objects++ allows you to use the native database vendor API, a common API/Method, or an abstract access method where no SQL knowledge is needed. SQL Objects++ supports OS/2 (32-bit), NT, Windows, and UNIX with the same API/Method. SQL Objects++ can be called from development tools that can call a C/C++ DLL. SQL Objects++ supports Oracle, Sybase, SQL Server, DB2/2, SQL Base, Watcom SQL, Btrieve, Netware SQL, Informix, and others. Price: \$695.

Objects++ Software Corp., 47 Stonewall St., Cartersville, Ga. 30120, (800) 876-6585 or (404) 382-6585, fax (404) 382-6374.

OBJECT TECHNOLOGY INTERNATIONAL INC.

Circle No. 129

ENVY/Developer is an integrated multiuser environment for serious Smalltalk development. It features a highly productive team programming environment that enables the development, release, and maintenance



nance of Smalltalk applications. Its version control and configuration management features provide control for software engineering while enhancing the individual Smalltalk developer.

Object Technology International Inc., 2670 Queensview Dr., Ottawa, Ont., Canada k2b 8K1, (613) 820-1200, fax (613) 820-1202.

ORANGE HILL SOFTWARE

Circle No. 130

IControl/DE 2.15 allows developers to choose from over 500 professionally drawn icons to use in programs or WPS objects. It is royalty free and provides the features of OHS' IControl Plus—automatic icon application, icon starter templates, developer's utilities, additional icons, and Workframe/2 integration.

Orange Hill Software, P.O. Box 907, Atwood, Calif. 92601, (714) 524-5851, fax (714) 996-7430.

PARCPLACE SYSTEMS INC.

Circle No. 131

VisualWorks 2.0 is a client/server tool for building portable applications using object-oriented technology. A Database Application Creator is included for developing applications that are portable across multiple platforms, scalable across the enterprise, and have their functionality distributed between both clients and servers. Price: \$429.95 for Windows, NT, Mac, OS/2; \$49.95 for UNIX.

ParcPlace Systems Inc., 999 East Arques Ave., Sunnyvale, Calif. 94086-4593, (408) 481-9090, fax (408) 481-9095.

PROTOSOFT

Circle No. 132

Paradigm Plus 2.0 is an object-oriented CASE tool that supports the industry's methodologies: Rumbaugh OMT, Shlaer/Mellor, Fusion, Booch, and Coad/Yourdon. Platform support includes PC/Windows, Sun OS, Solaris, HP-UX, and more. Other features include code generation in C, C++, Ada, Smalltalk, CORBA IDL, and SQL. Price: begins at \$3,995 for a PC fixed copy, \$7,700 for a UNIX copy; volume discount pricing schedule does apply.

Protosoft, 17629 El Camino Real, Suite 202, Houston, Tex. 77058, (713) 480-3233, fax (713) 480-6606.

POET SOFTWARE

Circle No. 133

Poet 2.1 is a C++ Object DBMS available in single user and client/server versions. Developers save time since there is no need for writing code to break up C++ objects into tables and to preserve their object's encapsulation, inheritance, identity, pointers, and polymorphism. Poet is portable and heterogeneous on OS/2, Windows, WFW, NT, Macintosh, Novell, and UNIX. Price: \$499 Personal Edition, \$3,495 Professional Edition.

Poet Software, 4633 Old Ironsides Dr. Ste. 110, Santa Clara, Calif. 95054, (800) 950-8845 or (408) 970-4640, fax (408) 970-4630.

SOFTBRIDGE INC.

Circle No. 134

Automated Test Facility 3.1 tests client/server and stand-alone object-

oriented OS/2 and Windows applications. From a central point, it runs simultaneous, unattended tests on multiple PCs. It tests all client/server layers: graphical user interfaces, distributed applications, and legacy systems. Users may take a capture/playback or full scripting approach to testing. Price: \$18,000 and up, depending on configuration.

Softbridge Inc., 125 Cambridge Park Dr., Cambridge Mass. 02140, (800) 955-9190 or (617) 576-2257, fax (617) 864-7747.

SYNTEGRATION

Circle No. 135

The Secure Workplace for OS/2 1.55 is a collection of OS/2 workplace objects and programs utilities designed to provide a secure desktop environment. These objects protect the OS/2 desktop from unwanted changes or intrusions. Objects can be set to a locked position. The product includes setup utilities to help you

Take control of your OS/2 scheduling needs with Advanced Task Scheduler for OS/2

September						
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

ATS

Version 3.0



Are you looking for a way to manage your production job streams? Do you have programs that need to run after the completion of one or more other programs? Do you want to run programs on a regular basis? Do you want to process files that have just been received from another system or modified locally? Do you need to take action if a file is missing? Do you need to schedule around holidays and periods of heavy CPU load?

With ATS you can run your programs when YOU want them to run with out you having to be there.

Upgrades Available
from Version 2 and
other Task Schedulers
Call or E-Mail for Details

Here are some of the features of ATS for OS/2:

- * Manage Production Job Streams
- * Programs can be scheduled to run anytime.
- * Programs can be dependent upon Files, Other Programs, Holiday Schedules, and External Signals
- * Integrated Operators Console
- * Logging - To File and Console

New Features:

- * Job Queues - for managing your resources while managing your tasks
- * Periodic Scheduling
- * COMPLETE API and Command Line Interface

Software and Consulting
2227 U.S. Highway #1 * Suite 146
North Brunswick, NJ 08902

Voice: (908) 821 - 0359 * Fax: (908) 821 - 0350 * CompuServe 70312,627

Only
\$349

Circle Reader Service Number 47



configure a secure workplace. Price: \$59.95 for single-station End User Edition, \$549 for 10-station Enterprise Edition.

Syntegration, 13241 Norton Ave., Chino, Calif. 91710, (909) 464-945, fax (909) 627-3541.

TECHNICALLY SPEAKING

Circle No. 136

Show N Tel lets you visualize and design applications of any size and scope. Slick graphical user interface provides unprecedented ease of use while providing extensive database support, host integration, high port density, error handling, on-line help, and a host of other features. Price: \$2,995.

Technically Speaking, 171 Main St., Ashland, Mass. 01721, (508) 881-1900, fax (508) 881-7320.

TOWER TECHNOLOGY CORP.

Circle No. 137

The TowerEiffel System 1.3 is a complete object-oriented programming system that includes a high-performance Eiffel 3 compiler, development environment, programming tools, and a base set of reusable software components. A unique capability of the TowerEiffel System is support for multilanguage development and interoperability between Eiffel, C, and C++. Price: \$1,295 commercial license, \$389 individual license,

\$129 university faculty license, \$100 60-day evaluation license.

The Eiffel Booch Components 1.2 is the Eiffel version of Booch Components. Reengineered to take advantage of Eiffel's unique capabilities, the Eiffel Booch Components are a carefully designed set of Eiffel clusters that provide a complete collection of efficient and adaptable domain-independent data structures and algorithms. Price: \$995 commercial license, \$299 individual license, \$99 university faculty license.

Tower Technology Corp., 1501 Koenig Ln., Austin, Tex. 78756, (800) 285-5124 or (512) 452-9455, fax (512) 452-1721.

TRINZIC CORP.

Circle No. 138

Aion Development System (AionDS) is a rule-based, object-oriented development tool for client/server and mainframe environments. AionDS is intended for problems that incorporate business policies and judgment. AionDS is typically used to develop applications that are too difficult to code and maintain productively in 3GL languages and 4GLs. Price: \$10,000.

Trinzic Corp., 101 University Ave., Palo Alto, Calif. 94301, (415) 328-9595, fax (415) 321-7728.

VISIX SOFTWARE

Circle No. 139

Galaxy 2.0 is a cross-platform devel-

opment environment designed for creating large-scale mission-critical applications that are graphical and distributed. Galaxy assembles a set of visual tools, object-oriented libraries, and distributed services into one environment. Applications can be built and compiled to run enterprise-wide on a broad range of desktop and server platforms, networks, and window systems without changing code. It can run with UNIX, Windows, Windows NT, Macintosh, OS/2, and Open VMS. Price: \$9,600 for C version, \$12,100 for C++.

Visix Software, 11440 Commerce Park Dr., Reston, Va. 22091, (800) 832-8668 or (703) 758-8230, fax (703) 758-0233.

ZINC SOFTWARE INC.

Circle No. 140

Zinc Application Framework 3.6 is an object-oriented, multiplatform, internationalized C++ class library. It supports IBM OS/2, Microsoft Windows, Win32s and Windows NTm OSF/Motif, Curse, DOS Graphics and DOS Text, Apple Macintosh, and NEXTSTEP within one set of source code. Zinc designer, a multiplatform visual development tool, is included. Price: \$798.

Zinc Software Inc., 405 South 100 East, 2nd Floor, Pleasant Grove, Utah 84602, (800) 638-8665 or (801) 785-8900, fax (801) 785-8996.



New Tools for OS/2



Review. This image archival, storage, and retrieval system has a hierarchical file system, direct scanner support, file format support, and file conversion for importing images. It currently supports the HP IIP, IIC, and IICX scanners and will be able to support other scanners later this year.

In a separate event, the company has recently announced a maintenance and feature release of APPLAUSE 1.1. It includes support for Photo CD, IBM IOCA, Targa, and others. It will be sent to all registered users.

Solution Technology Inc. Circle No. 141
Phone: (407) 241-3210, Fax: (407) 997-6518

SYSQL For Windows. This middleware product takes advantage of the 32-bit architecture of OS/2 and provides access to S/390 data and services from OS/2 applications. Users may develop client/server applications on OS/2 platforms using any development tool (like C, COBOL, REXX, and CA's Realizer) and get direct access to data and services (such as VSAM, DB2 or ADABAS files, JES Spool files, and PDS libraries) on Host MVS systems using SLYSQL facilities. SYSQL also supports Remote Procedure Calls and comes with more than 50 predefined mainframe functions, to which users can add their own.

Text Systems International Circle No. 142
Phone: (203) 637-4549, Fax: (203) 698-2409

PM Assistant 2.0. This suite provides hot-key access to applications, automatic window placement, keyboard macros, task scheduling, a screen saver, and password protection. The six utilities are integrated into a single seamless program and can be used together to accomplish complex tasks. PM Assistant 2.0 runs on OS/2 2.0 or greater and can automate and customize OS/2 and applications according to each user's preferences. PM Assistant does not contain a programming language; all utilities are configured with dialog boxes and by pointing to the application to act on.

Utilis Inc. Circle No. 143
Phone: (206) 467-4025

Quantum Leap 2.1. This is an advanced version of Quantum Leap for OS/2 business problem solving software. It capitalizes on the new symmetric multiprocessing (SMP) version of OS/2, providing the ability to "scale up" to SMP machines and take advantage of a heterogeneous network of personal computers and RISC System/6000 workstations. Quantum Leap 2.1 provides Application Programmer Interfaces (APIs) that allow consultants, corporate developers, and value-added resellers to embed Quantum Leap models and problem-solving technology in solutions created for their users. Users can create and maintain complex models, representing dynamics of a real-life business environment by using the graphical user interface. Training and consulting services are offered through Quantum Craft to support many industries.

Quantum Development Corp. Circle No. 144
Phone: (302) 798-0899, Fax: (302) 798-6813

AM COBOL. This software for OS/2 allows developers to reuse existing COBOL code in client/server applications and write portions of client/server applications in COBOL. The AM COBOL Interface links client/server applications with background Micro Focus COBOL programs in a sound peer-to-peer architecture. AM COBOL Interface helps corporate programmers with COBOL backgrounds use their skills for client/server development. According to the company, AM customers can better leverage their 3GL programming skills.

The company has also announced the release of AM AAI Interface, AM Interface for CICS OS/2, and AM Interface for Existing DLLs.

Intelligent Environments Circle No. 145
Phone: (508) 640-1080, Fax: (508) 640-1090

Notrix. Percussion has released a programming tool for easily manipulating large, complex Notes data-



bases and building Notes workflow applications. Notrix fills the "programming gap" between the Lotus Notes macro language and the Lotus Notes application programming interface.

Notrix Composer Version 1 for EDA/SQL. This tool lets Notes users view and manipulate data from more than 50 different databases supported by Information Builders' Enterprise Data Architecture (EDA)/SQL client/server software.

Both products include the Notrix Event Manager, which acts as an active information agent, enabling customers to automatically trigger events within Notes or in outside applications in response to designated events within Lotus Notes databases, documents, and document fields. Both products run on OS/2-based servers and work with any Notes-supported client, including OS/2, Windows, and Macintosh without additional interface programming or client software.

Percussion
Software Inc. Circle No. 146
Phone: (617) 267-6700, Fax: (617) 266-2810

Unite Imaging Object Set. This object set, based on IBM's SOM technology, is designed for mission-critical OS/2 applications and includes Image Viewer, Scanner Manager, and Storage Manager. IBM's SOM object-oriented programming interface is language independent and supported by OpenDoc, Taligent, and OS/2 for the Power PC. SOM methodology allows the Cirrus objects to be accessed not only from C++ but from procedural languages as well. The object set is available as fully functional APIs or can be integrated as enabled parts with IBM's VisualAge development environment or with Watcom's VX-REXX development environment. Users can implement a common set of reusable components in applications to reduce development time. The object set is designed for OS/2 networks.

Cirrus Technology Inc. Circle No. 147
Phone: (301) 698-1900

CasePoint. The company has announced an OS/2 tool selected by IBM for inclusion as a primary tool in enhancing support for OS/2 in a new companion service offering for problem resolution called

the Technical Connection CD-ROM, available directly from IBM. CasePoint helps businesses reengineer customer support with problem guidance and resolution. CasePoint provides guided access to an evolving case of experience, such as prior customer problems with known solutions. The user interface accepts user input in freeform natural language, proposes clarifying questions to ask the caller, interprets the answers, and identifies those cases that explain and resolve the caller's problem.

Inference Corp. Circle No. 148
Phone: (310) 322-0200, Fax: (310) 322-3242

Just Logic/SQL Database Manager. The relational database server is made to be used from C and C++ programs and is meant for professionals who write simple to complex C applications and need a flexible performing database. There are three programming interfaces to choose from: a complete set of C++ class definitions, a complete C Application Program Interface, and a standard C precompiler. The database delivers a true relational database on OS/2 and compatibility with most popular compilers.

Just Logic
Technologies Inc. Circle No. 149
Phone: (800) 267-6887 or (514) 761-6887,
Fax: (514) 642-6480

Thompson Toolkit 3.0. The toolkit provides full compatibility, which allows you to customize UNIX or DOS to your preference. It has a new network support server, a command history window, and a "vi"-compatible editing mode. Other

features include an "exec" command for shell I/O redirection, a "select" command to automatically print menus, a "dosenv" command to modify the DOS environment, a "fullpath" command to construct absolute path names, and a "[[]]" command to test variables, files, and options. The product has a random number generator, lapsed time counter, and new options in SHEXPAND variable to provide greater control of DOS command line expansion.

Thompson
Automation Inc. Circle No. 150
Phone: (800) 944-0139 or (503) 224-1639,
Fax: (503) 224-3230

CC-Rider. The standard edition of the package includes both the CCSYM Symbol Compiler and the CCRider pop-up utility, providing basic analysis for small or large projects. It includes function prototyping, log file generation, QuickHelp support, and hypertext retrieval of all symbol definitions from the editor. The professional edition provides more extensive source code analysis for big projects and can cross reference all active symbol usages. The editor can navigate through source files with knowledge of a program's functions and variables.

Western Wares Circle No. 151
Phone: (303) 327-4898

SMART 2.0. An enhanced version of the migration toolset that shipped in May, this version adds functionality that will double the speed of a conversion effort. SMART 2.0 supports 16-bit and 32-bit Windows application migration to 32-bit OS/2 as well as 16-bit OS/2 to 32-bit OS/2. It

OS/2 CALENDAR

Oct. 4-6	Software Development East '94, Washington, D.C. (415) 905-2414
Oct. 30-Nov. 4	ColoradOS/2, Colorado Springs, Colo. (719) 576-5003
May 8-10, 1995	OS/2 Development '95 Amsterdam, Holland Phone: 32-3-358-6064 Internet: csp@mfi.com Compuserve: 71673,1163

includes an analysis and reporting tool and a source migration tool. The first sizes a conversion effort and provides a road map for the migration, thereby minimizing work effort. The second changes up to 70 percent of the API and message code and provides reports to help programmers migrate the remainder of the code.

One Up Corp. Circle No. 152
Phone: (214) 620-1123, Fax: (214) 620-9626

Object COBOL Option. An add-on to the Micro Focus COBOL Workbench 3.2, this product provides developers with a comprehensive object-oriented development environment and class library and enables the development of graphical user interface applications using Object COBOL. The option allows application developers to migrate existing code to object orientation incrementally because it allows object

and procedural code to coexist and cooperate. Because Object COBOL syntax is processed by the COBOL compiler (not the preprocessor), the Micro Focus Object COBOL Option debugs code at the source level to increase programmer efficiency and reduce error.

Micro Focus Circle No. 153
Phone: (415) 856-4161

NDP OS/2 Developer's Pack. Microway is now shipping four compilers for OS/2: NDP Fortran-77, NDP Fortran-90, NDP C/C++, and NDP Pascal. These products take advantage of the advanced features of the Intel 386/486 architecture along with OS/2's ability to load and run 32-bit applications on the 386, 486, and i860. Mainframe libraries like NAG, IMSL, and Kuck are also available for the NDP OS/2 compilers; the compilers include MGX,

Microway's device-independent, vector-based graphics and plotting package.

Microway Circle No. 154
Phone: (508) 746-7341, Fax: (508) 746-4678

PCYACC 4.0. This language development environment generates ANSI C source code from input language description grammars for building assemblers, compilers, interpreters, browsers, page-description languages, language translators, syntax-directed editors, language validators, natural language processors, expert system shells, and query languages. It includes 32-bit extended memory support for OS/2 2.0 that will compile any size grammar, debugging tools, and new C++ and YACC classes for OS/2 2.0 PM.

Abraxas Software Circle No. 155
Phone: (503) 244-5253, Fax: (503) 244-8375



Multiple protocols. One slot.



Use Quadron software to save PC expansion slot space and money by running two or more protocols on a single co-processor card. Each port can have a different protocol and line speed.

**Async • Bisync • HDLC/SDLC
X.25 • LAP-B • Custom protocols**

We currently support OS/2, DOS, UNIX and AIX. With little or no modification, you can port your applications from one to another. Call today to find out how we can help you communicate better.



Quadron

805-966-6424

209 East Victoria Street, Santa Barbara, CA 93101

We'd like to dispel a couple of
myths about client/server testing.

Myth #1: Client/Server testing is easy.

New development tools make building client/server applications easier than ever. The problem is, they're harder to test, and you just can't stand up to them with manual testing methods. **You need ATF, the only testing product designed for client/server.**

Myth #2: Client/Server testing is impossible.

Testing client/server applications can be daunting. The problem is, your tests need to replicate real-world conditions, and you just can't do that with a stand-alone testing product. **You need ATF, the only testing product designed for client/server.**

The Softbridge Automated Test Facility's unique architecture makes it the only product for true client/server testing of OS/2 and Windows apps. To learn more about ATF, call 617-576-2257. (Or FAX 617-864-7747.)

If you're coming to *OS/2 World* in Santa Clara this July, stop by Booth 615 for an ATF demo. **There's only one way to test client/server applications, and that's with ATF.**

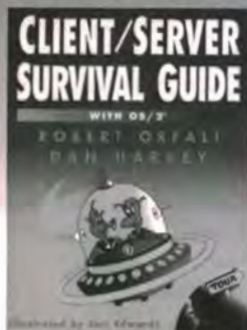


Softbridge, Inc. ▲ 125 CambridgeParkDrive ▲ Cambridge, MA 02140

Circle Reader Service Number 48

Circle Reader Service Number 44

VNR KNOWS OS/2®



CLIENT/SERVER SURVIVAL GUIDE WITH OS/2 2.1 by Robert Orfali and Dan Harkey. A sweeping tour of client/server systems and software. \$39.95 0-442-01798-7 IBM# SR28-5494

CLIENT/SERVER PROGRAMMING WITH OS/2 2.1, 3rd Ed., by Robert Orfali and Daniel Harkey. "Absolutely one of the most impressive, useful, best-written computer books I've ever seen."—Will Zachman. \$39.95 0-442-01833-9 IBM# G325-0650-02

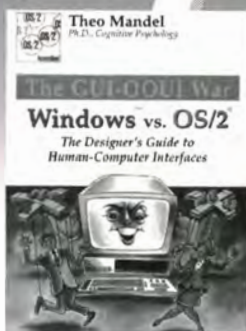
THE GUI-OOUI WAR: Windows vs. OS/2 by Theo S. Mandel. "A book that should be forced on every Windows and OS/2 developer working today. An exceptionally clear, readable guide. If you want to make a good program great, there's no better starting place."—OS/2 Professional. \$29.95 0-442-01750-2 IBM# SR28-5251

OS/2 FUNCTIONS QUICK REFERENCE LIBRARY 1: Win Functions by Nora Scholin. \$19.95 0-442-01897-5

C AND C++ PROGRAMMING IN THE OS/2 ENVIRONMENT by Mitra Gopaul. \$39.95 0-442-01240-3 IBM# SR28-4404

OS/2 2.1 APPLICATION PROGRAMMER'S GUIDE by Jody Kelly, Craig Swearingen, Dawn Bezviner and Theodore Shrader. \$34.95 0-442-01736-7 IBM# SR28-5449

OS/2 2.1 REXX HANDBOOK: Basics, Applications, and Tips by Hallet German. A great reference book that contains info not found in any other REXX books. \$29.95 0-442-01734-0 IBM# SR-5250



THE OS/2 2.1 CORPORATE PROGRAMMER'S HANDBOOK by Nora Scholin, Mark Sullivan, and Robin Scragg. "A solid job. Introduces readers to programming terms and concepts. Explanations are lucid and reading relevant sections provides you with the basics you need."—OS/2 Professional. Includes disk. \$39.95 0-442-01598-4 IBM# SR28-4390

WRITING OS/2 2.1 DEVICE DRIVERS IN C, 2nd Ed., by Steven Mastrianni. "Should give any C programmer with a vague knowledge of OS/2 and PC hardware a headstart in writing OS/2 device drivers in C."—International OS/2 User Group Magazine. Includes disk. \$39.95 0-442-01729-4 IBM# SR28-4392

THE SHELL COLLECTION: OS/2 2.X Utilities by Steven Levenson. A great source of shortcuts, tips and productivity boosters for Windows and DOS. Includes disk. \$29.95 0-442-01585-2 IBM# G362-0049

USING WORKPLACE OS/2: Power User's Guide to IBM OS/2 Version 2.1 by Lori Brown and Jeff Howard. The authors are the principal designers of the Workplace Shell! Includes disk. \$24.95 0-442-01590-9 IBM# SR28-4391

THE OS/2 2.0 HANDBOOK: Applications, Integration, and Optimization by William H. Zack. "Should be within fingertip reach of anyone working with OS/2. IBM would do well to include a copy in every package of OS/2."—OS/2 Professional. \$34.95 0-442-01234-9 IBM# G362-0009

OS/2® is a registered trademark of the IBM Corporation.



OS/2 2.X NOTEBOOK: The Best of IBM OS/2 Developer, edited by Dick Conklin. \$36.95 0-442-01522-4 IBM# G362-0015

NOW THAT I HAVE OS/2 2.1 ON MY COMPUTER, WHAT DO I DO NEXT?, 2/E. by Steven Levenson. "Easy to read, loaded with examples, a friendly reference source."—Journal of The Chicago Computing Society. \$22.95 0-442-01832-0 IBM# G362-0008-01

COMPREHENSIVE DATABASE PERFORMANCE FOR OS/2 2.0's Extended Services by Bruce Tate, Tim Malkemus and Terry Gray. \$41.95 0-442-01325-6 IBM# G362-0019

LEARNING TO PROGRAM OS/2 2.0 PRESENTATION MANAGER BY EXAMPLE: Putting the Pieces Together by Stephen Knight. Includes disk. \$41.95 0-442-01292-6 IBM# G362-0011

THE COBOL PRESENTATION MANAGER PROGRAMMING GUIDE by David M. Dill. \$41.95 0-442-01293-4 IBM# G362-0010

OS/2 PRESENTATION MANAGER GPI GRAPHICS by Graham C. E. Winn. \$42.95 0-442-00739-6 IBM# G362-0005



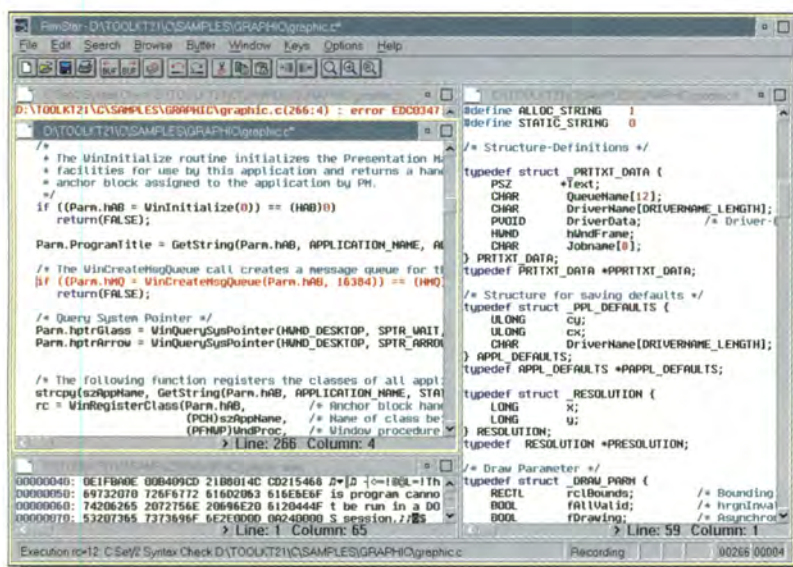
Order directly from the publisher by calling **1-800-544-0550**. To order from IBM call 1-800-568-2694

Van Nostrand Reinhold

115 Fifth Avenue, New York, New York 10003

1-800-544-0550 73373.64 @compuserve.com

POWER UP WITH THE RIMSTAR™ PROGRAMMER'S EDITOR NEW VERSION 2.1



If you're looking for a fully configurable, professional OS/2 PM programmer's editor to replace your current text-mode editor – we have what you have been looking for!

No hassle changing editors

We know changing editors can be a major hassle. You don't want to relearn a new set of keystrokes no matter how good the product. Well, you don't have to – we have Brief, CUA, Epsilon, Multi-Edit, SlickEdit, PWB, and Borland IDE keyboard mapping waiting for you. If you're not using one of these, let us know and we will create the mappings you need.

Features designed to increase productivity

It has all the features you have come to expect, plus special features designed to anticipate your needs and increase your productivity. Features like a 'C' source browser that eliminates those time consuming searches for functions, global variables, typedefs, and macros by providing you with instant positioning to both definitions and references. Using our powerful 'C' macro language you can customize your programming environment to suit your individual needs.

RimStar Technology, Inc.

91 Halls Mill Road
Newfields, NH 03856
Voice: (603) 778-2500
Fax: (603) 778-2408
BBS: (603) 778-4644

Price \$299.00

Plus Shipping & Handling.

To order call 1-800-746-7007
60 day money-back guarantee.

Also available for Windows and Windows NT

Circle Reader Service Number 51

Partial Feature List

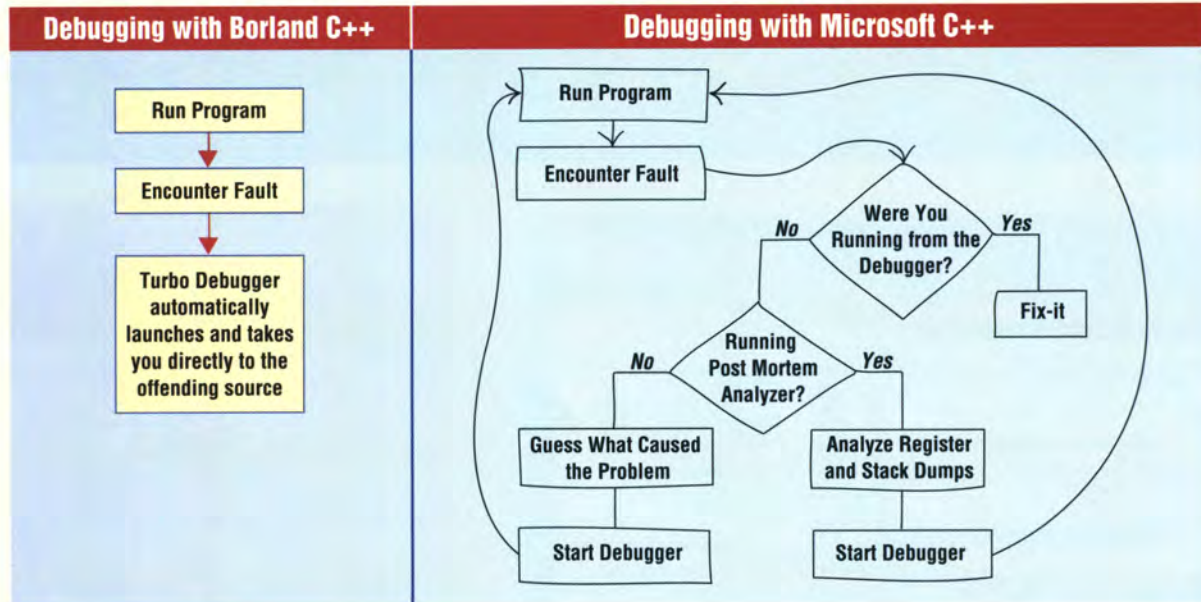
- ✓ Complete ANSI 'C' macro language
- ✓ SyntaColor™—syntax highlighting
- ✓ Smart C/C++ indenting & brace matching
- ✓ Bookmarks
- ✓ Unlimited undo and redo
- ✓ Timed auto save
- ✓ Access PDK help for function under cursor
- ✓ Multi-threaded for no waiting
- ✓ Compile and jump to errors
- ✓ Import/export to system clipboard
- ✓ Keystroke record/playback
- ✓ Block indent/outdent
- ✓ Hex editing
- ✓ Customizable menus
- ✓ Column, line and block selection, search and replace
- ✓ Integrates with Workframe/2
- ✓ Source browser for 'C'
- ✓ Template editing
- ✓ Multi-buffer regular expression search and replace
- ✓ Support for version control
- ✓ Complete on-line help
- ✓ Save and restore state between sessions
- ✓ OS/2 2.x 32 bit PM Multi-document interface

"My copy of Brief has been permanently retired. Keep up the good work!" - A.L.

All products and company names are trademarks or registered trademarks of their respective holders. RimStar and SyntaColor are trademarks of RimStar Technology, Inc.

© 1994 RimStar Technology Inc.

Borland is first to deliver "Just-in-Time" debugging



The Fastest Way to Find and Fix Bugs

You're running your program. You hit an edge condition. You crash. Unexpected? Of course. Chances are you weren't running your debugger. What happens next? It depends upon the tools you use. If you're using Borland® C++, Turbo Debugger will automatically launch and take you to your problem. If you're using any other C++, you may be in for a long ride home.

Only Borland has Just-in-Time Debugging

The Turbo Debugger, in Borland C++, is now better than ever. Now you can debug your application even if the debugger isn't running. A hot new Borland technology in the Turbo Debugger called "Just-in-Time Debugging" makes it possible. With Just-in-Time Debugging, if you run into a fault under NT or Chicago, Turbo Debugger will automatically load and take you directly to the offending piece of code. That means you can debug your program even if you hadn't already started the debugger. You catch and fix bugs faster...you ship sooner with Borland C++.

Only Borland has Exception Handling and RTTI

Of course preventing bugs is better than fixing them. That's why Borland is at the forefront in delivering new language features that not only save you time in debugging, they can actually save your project. Features like runtime type information prevent improper casts that can easily cause your program to exhibit unpredictable behavior. And exception handling which lets you recover cleanly from crashes that otherwise would leave objects on the stack—or even worse, leave your customer's data in unstable states.

Get Borland C++

For years, Borland's Turbo Debugger has been saving development projects. With conditional breakpoints, expression evaluation, remote capabilities, exception handling support, and

Buy Borland C++
and get a **FREE**
Training Course
Courses available for C++
Training Course at your local reseller

Borland
The Upsizing Company

now, Just-in-Time Debugging and DPAPI support. Buy Borland C++ today and put Just-in-Time Debugging to work for you. For more information call: **1-800-645-4559, ext. 8842**

Subscribe Now to Our Free Technical Newsletter: Send Mail to tech-info@borland.com